

W OBRONIE WOLNOŚCI

Tytuł oryginału: Free as in Freedom

Tłumaczenie: Krzysztof Masłowski

(c) 2003 Wydawnictwo Helion.

Authorized translation of the English edition (c) 2002 Sam Williams.

This translation is published and sold by permission of O'Reilly & Associates, Inc.,
the owner of all rights to publish and sell the same.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being no invariant sections, with the Front-Cover Texts being no invariant sections, and with the Back-Cover texts being no invariant sections. A copy of the license is included in Appendix C, GNU Free Documentation License. All images are to be included verbatim when the document is copied, distributed, or modified under the terms of the GFDL.

Polish language edition published by Wydawnictwo Helion.
Copyright (c) 2003

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Wydawnictwo HELION dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Wydawnictwo

HELION nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Wydawnictwo HELION

ul. Chopina 6, 44-100 GLIWICE

tel. (32) 231-22-19, (32) 230-98-63

e-mail: helion@helion.pl

WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie?wobron>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Mojej żonie Tracy

PODZIĘKOWANIA

Specjalne podziękowania należą się Henningowi Gutmannowi za zainspirowanie mnie do napisania tej książki. Dziękuję także Baronowi Oasowi za uprzednie zasugerowanie tego pomysłu Tracy. Dziękuję Laurie Petrycki, Jeffreyowi Holcombowi i pozostałym osobom z O'Reilly & Associates, a szczególnie Timowi O' Kelly za poparcie udzielone tej książce. Wyrazy uznania należą się grupie osób sprawdzających tekst, a byli to: Bruce Perens, Eric Raymond, Eric Allman, Jon Orwant, Julie i Gerard Jay Sussman, Hal Abelson oraz Guy Steele. Mam nadzieję, że obecna wolna od błędów wersja dała Wam wiele

radości. Dziękuję również Alice Lippman za wywiady, ciasteczka i fotografie. Wreszcie moja rodzina: Steve, Jane, Tish i Dave - im składam równie gorące podziękowania. Na koniec wyrażam wdzięczność Richardowi Stallmanowi za to, że nie zabrakło mu serca ani cierpliwości, gdy "pokazywał mi kod".

- Sam Williams

WSTĘP

Działalność Richarda M. Stallmana mówi sama za siebie: od udokumentowanych kodów źródłowych, przez liczne publikacje, po zarejestrowane rozmowy i wywiady. Niewiele jest osób, które tyle wysiłku wkładają w przekazanie innym swoich myśli i przekonań.

Taka otwartość - użyjmy na chwilę tego niestallmanowskiego określenia - jest niecodzienna. Żyjemy przecież w społeczeństwie, które traktuje informację, a zwłaszcza informację osobistą, jako towar. Zatem rodzi się pytanie, dlaczego istnieje ktoś, kto chce dzielić się z innymi taką masą informacji, nie żądając nic w zamian.

Jak przekonamy się w kolejnych rozdziałach, poczynania Stallmana nie są w pełni altruistyczne. Każdy program, przemówienie i zapisany bon mot ma przypisaną cenę, która jednak nie określa płatności w znaczeniu, do jakiego ludzie przywykli.

Nie jest to ostrzeżenie lecz zwykłe stwierdzenie faktu. Jako ktoś, kto spędził ostatni rok na wygrzebywaniu faktów z życia Stallmana, mogę się obawiać wystąpienia z krytyką jego dorobku. "Nigdy nie podejmuj walki z kimś, kto kupuje atrament beczkami" - mawiał Mark Twain. Trudno się pokusić o napisanie wyczerpującej biografii kogoś takiego jak Stallman, który każdą swą myśl podaje do publicznej wiadomości. Czytelników, którzy zdecydują się poświęcić kilka godzin na

przewertowanie tej książki, zapewniam, że znajdą tu fakty i cytaty, których nie da się odszukać na stronach internetowych za pomocą Google lub innej wyszukiwarki. Ale dostęp do tych informacji kosztuje. Kupując książkę, płacicie zań w tradycyjny sposób, zaś w przypadku wersji elektronicznej opłata jest pobierana jak za wolne oprogramowanie¹. Dzięki wydawnictwu O'Reilly & Associates książka ta jest rozprowadzana na zasadzie Licencji Wolnej Dokumentacji GNU², co oznacza, że każdy może jej tekst poprawiać, tworzyć własną wersję i upowszechniać na zasadzie tej samej licencji.

Jeżeli czytacie wersję elektroniczną i wybieracie drugi rodzaj płatności, tzn. chcecie przyczynić się do poprawienia treści ku pożytkowi następnych czytelników, czekam na Wasze teksty.

Wersję w formacie html publikuję od czerwca 2002 r. pod adresem internetowym <http://www.faifzilla.org/>. Celem jest regularne uaktualnianie i rozszerzanie tekstu Free as In Freedom. Jeżeli zamierzacie wziąć udział w tym przedsięwzięciu, czyli wybieracie drugą formę płatności, przeczytajcie dodatek C, gdzie został ypodane Wasze prawa wynikające z Licencji Wolnej Dokumentacji GNU.

I jeszcze uwaga końcowa: książka, którą trzymacie w ręku, to robota dziennikarska lecz również dokumentalna. Podczas jej pisanie i przygotowywania do druku korzystałem z licznych uwag i komentarzy wielu uczestników opisywanych wydarzeń, łącznie z samym Richardem Stallmanem. Uświadomiliśmy sobie wówczas, że w tekście zostało zawarte wiele informacji technicznych, których dokładniejsze lub lepsze wyjaśnienie bardzo ułatwiłoby lekturę. Ponieważ książka została awydana na zasadach Licencji Wolnej Dokumentacji GNU, akceptujemy wszelkie poprawki, tak jakbyśmy robili w przypadku opublikowania wolnego oprogramowania. Zaakceptowane zmiany zostaną łączone do wersji elektronicznej oraz, być może, także do przyszłejwersji drukowanej. Jeżeli chcecie się przyczynić do podniesienia jakości i wartości tej książki, możecie przesłać mi informację na adres sam@inow.com.

Przypisy

1 Termin angielski "free software" bywa tłumaczony różnie; zwykle jako "wolne oprogramowanie" lub "bezpłatne oprogramowanie". Istnieje Free Software Foundation, na której polskiej stronie pod adresem <http://www.gnu.org/home.pl.html> znajdujemy nazwę "Fundacja Wolnego Oprogramowania", zaś "Słownik terminów komputerowych" wydawnictwa Prószyński i S-ka nazywa ją "Fundacją na rzecz Bezpłatnego Oprogramowania". Sądząc, że twórcy strony fundacji lepiej wiedzą, jak się ich organizacja nazywa, przyjmuję nazwę "wolne oprogramowanie", tym bardziej, że ta forma jest bliższa wersji angielskiej - przyp. tłum.

2 GNU Free Documentation Licence. GNU to rekurencyjny akronim zdania „GNU's Not Unix”, czyli „GNU to nie Unix”. Więcej informacji można znaleźć pod adresem WWW podanym w poprzednim przypisie - przyp. tłum.

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 maszyny 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 odarem 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 publishing³, które do końca dekady miały zająć znaczące miejsce na rynku komputerowym.

Nagłymi 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. Zadrukowane kartki opuszczała 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 były okrągłe, a nie owalne, linie proste były naprawdę liniami prostymi, a nie sinusoidami o niewielkiej amplitudzie. Był to ze względu na to, że należało docenić. Jednakże po kilku tygodniach zaczął wydawać o sobie znać słabe punkty urządzenia. Głównym była skłonność do zawieszania się wskutek zaklinowania papieru. Programiści o zdolnościach inżynierskich wkrótce zrozumieli przyczynę. Urządzenie, 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 miał 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 wysłał aną 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 był owykorzystaniem 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 zjawią się tam dwie lub trzy zainteresowane osoby i zajmą się problemem. Z tych dwóch lub trzech osób przynajmniej jedna będzie wiedział a, 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 niejawnny zawierał a 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 darował 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 zwyższającym 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 nieznany 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, musiał oto 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 użyczenie 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 żądaniami?"

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 World 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 wyłączeniowy "podłożone ładunki".

Dla Reida była to procedura doskonała. Scribe nie stał się publicznie dostępny, a Unilogic odzyskał zainwestowane pieniądze. Natomiast dla Stallmana był 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 źródeł owych oprogramowania drukarki Xeroxa odbijały się od muru zмовy milczenia. Wreszcie pojawił się, 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. Przedstawivszy 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 dotarł do niego prośba Stallmana, był w Carnegie Mellon zapewne niecałym miesiącem".

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, był 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 dotychczas 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 dyskrecji. 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, przemienił y 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 stanowił yone 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 miał o 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 przebudzenia się całego przemysłu oprogramistycznego.

Oprogramowanie stało się dla firm rzeczą tak ważną, że uznali za publikowanie kodów źródłowych za zbędne i niebezpieczne, gdyż mógł zostać 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 był o 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, wywołała nieufność, z jaką od tej pory podchodził do wszystkich darów, a po drugie, postawił o 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 doprowadził odo wniosku, że umowy tego typu muszą mieć swoje ofiary" - twierdzi z przekonaniem Stallman - "W tym przypadku ofiarą był em 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 - "Odpowiedział em zatem: Ździękuje 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 hakera". 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 pobięł dby 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

byłaby też uzasadnionej złości i emocji, które wkrótce stały się siłą napędową jego działań, na podobieństwo ideologii politycznej lub wiary w zasady etyczne.

"Tego dnia pojawiłem 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 powstał w 1895 roku z inicjatywy kilkunastu 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 formatowania 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 Uczelnia powstała w Pittsburgu w roku 1900 z fundacji

przemysłowca i filantropa Andrew Carnegiego. W ciągu ponad 100 lat istnienia trzykrotnie zmieniał 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 doskonałej 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.

2. 2001: ODYSEJA HAKERSKA

Wydział nauk komputerowych New York University (NYU) 1 znajduje się w Warren Weaver Hall, przypominającym fortecę gmaszysku oddalonym o dwa kwartały na wschód od Washington Square Park. Wyloty instalacji klimatyzacyjnych dużej mocy tworzą wokół niego wał gorącego powietrza zniechęcający w równej mierze łazków, jak i radców prawnych. Przełamawszy barierę gorącego powietrza, odwiedzający trafiają na następną, równie nieprzyjemną - kontrolę straży pilnującej jedyne wejście.

Za punktem kontrolnym atmosfera łagodnieje. Nadal jednak liczne znaki rozsiane na pierwszym piętrze przypominają o zagrożeniu ze strony niezabezpieczonych drzwi i

niezaplombowanych wyjść awaryjnych. Nawet w niezmałconej atmosferze Nowego Jorku sprzed 11 września 2001 roku wszystko to uświadamiało, że nikt nie jest zbyt bezpieczny lub zbyt podejrzliwy.

Znaki te stanowiły tematyczny kontrast dla gęstniejącego tłumy gości zbierających się w wewnętrznym atrium. Nieliczni wyglądali na studentów NYU. Większość przypominała a raczej rozchętanych fanów muzyki współczesnej tłoczących się przed salą koncertową w oczekiwaniu na główny punkt programu. Na jedno krótkie przedpołudnie masy te zawładnęły Warren Weaver Hall, ograniczając zadania pilnującej bezpieczeństwa portierki do oglądania telewizji i wskazywania ruchem ramienia audytorium, w którym odbywa się "przemówienie".

Wewnątrz audytorium przybywający mogli zobaczyć osobę, która spowodowała czasowe zawieszenie zwykłych procedur bezpieczeństwa. Był to Richard M. Stallman, twórca projektu GNU, prezes Fundacji Wolnego Oprogramowania², zdobywca MacArthur Fellowship³ w 1990, Association of Computing Machinery's Grace Murray Hopper Award⁴ (także w roku 1990), współzdobycza Takeda Award⁵ za rok 2001 i był yhaker z AI Lab. Zapowiadany jako gospodarz "okołohackerskich" stron internetowych, w tym <http://www.gnu.org>, Stallman zjawił się na Manhattanie, gdzie się urodził, by wygłosić przemówienie o łatwej do przewidzenia treści, w którym wystąpił przeciwko prowadzonej przez Microsoft kampanii zwalczania projektu Powszechnej Licencji Publicznej GNU⁶.

Tematem wykładu Stallmana była historia i przyszłość ruchu wolnego oprogramowania. Istotne jest miejsce jego wygłoszenia. Mniej niż miesiąc wcześniej w należącej również do NYU, położonej nieopodal Stern School of Business przemawiał Craig Mundie, jeden z prezesów Microsoft Corporation, grzmiąc i odsadzając od czci i wiary Powszechną Licencję Publiczną GNU wymyśloną przed Stallmana przed 16 laty, po perypetiach z drukarką laserową Xeroxa w r. 1980. Stworzona w celu przeciwdziałania narastającej fali utajniania oprogramowania,

GPL ewoluowała, by stać się głównym narzędziem społeczności zwolenników wolnego oprogramowania. Mówiąc prościej, GPL nadaje programom komputerowym status własności społecznej lub, jak kto woli, powszechnej z zachowaniem legalności praw autorskich ((c) copyright). Nadanie tych praw jest nieusuwalne. Wszystkie wersje pochodne podlegają tej samej zasadzie ochrony praw autorskich, nawet te, które zawierają jedynie niewielki wycinek oryginalnego kodu źródłowego. Z tego powodu licencja ta w świecie przemysłu programistycznego została nazwana "wirusową", gdyż zarażał każdy program, którego dotknęła⁷.

W biznesie informacyjnym, coraz bardziej zależnym od standardów oprogramowania, do GPL przywarła nazwa straszaka. Nawet firmy, które z początku z lekceważeniem naśmiewały się z niej, traktując ją jako programistyczną odmianę socjalizmu, zaczęły wolno zmieniać zdanie i zauważać niesione przez nią korzyści. Linux - uniksopodobne jądro systemu stworzone przez fińskiego studenta Linusa Torvaldsa w roku 1991 - jest rozpowszechniany z licencją GPL, tak samo jak wiele powszechnie w świecie stosowanych narzędzi: GNU Emacs, GNU Debugger, GNU C Compiler itd. Razem narzędzia te tworzą wolny system operacyjny zbudowany przez hakerów i będący własnością ich światowej społeczności. Zamiast uważać tę społeczność za groźbę firmy najnowszej technologii, jak IBM, Hewlett Packard i Sun Microsystems, zaczęły polegać na jej osiągnięciach i sprzedawać aplikacje i usługi, korzystając ze wciąż rosnących zasobów wolnego oprogramowania.

Jako broń strategiczną wykorzystują również wieczną wojnę społeczności hakerskiej z Microsoft Corporation, firmą programistyczną z Redmond w stanie Washington, która w latach osiemdziesiątych zdominowała rynek oprogramowania komputerów PC. Jako właściciel popularnego systemu operacyjnego Windows, Microsoft ma najwięcej do stracenia w przypadku przesuwania się światowego przemysłu w stronę

oprogramowania z licencją GPL. Niemal każda linia kodu źródłowego Windows jest chroniona prawami autorskimi potwierdzającymi prywatną naturę własności lub przynajmniej prywatne prawa Microsoftu do korzystania z tego oprogramowania. Z punktu widzenia Microsoftu włączenie do Windows programów na licencji GNU byłoby podobne do zażywania tabletek poronnych zmniejszających populację własnych programów i klientów. Konkurencyjne firmy zdobyłyby nagle prawo kopiowania, modyfikowania i sprzedawania poprawionych wersji Windows, podkopując pozycję Microsoftu jako firmy numer 1 na rynku oprogramowania, dostarczającej wciąż nowe wersje pełnego pakietu i błędów systemu. Zatem wiele firm z coraz większym zainteresowaniem zaczęło się zastanawiać nad możliwością "adopcji" oprogramowania GPL. Z tego powodu Mordy w swoim przemówieniu wyklinał oprogramowanie z "ogólnie dostępnymi kodami źródłowymi", jego rozwój i rosnącą sprzedaż. Dlatego Stallman postanowił publicznie odpowiedzieć na te zarzuty w tym samym kampusie uniwersyteckim.

20 lat to dla przemysłu programistycznego bardzo długi okres. W roku 1980, gdy Stallman w AI Lab miał kłopoty z drukarką laserową Xeroxa, Microsoft, postrzegany obecnie przez hakerów jako główne zagrożenie, dopiero raczkował jako niewielka prywatna firma. IBM, wówczas najpotężniejsza firma przemysłu komputerowego na świecie, wprowadził właśnie na rynek swój pierwszy komputer osobisty, dając początek tworzeniu rynku taniego sprzętu komputerowego. Wiele technologii, których istnienie przyjmujemy teraz za rzecz oczywistą: Word Wide Web, telewizja satelitarna, 32-bitowe konsole gier wideo, w ogóle wówczas nie istniało. To samo dotyczy wielu firm znajdujących się obecnie w czołówce establishmentu, jak np. AOL, Sun Microsystems, Amazon.com i Dell. To wyliczanie można długo kontynuować.

Fakt, że rynek wysokich technologii rozwinął się tak bardzo w tak krótkim czasie, przyczynił się do wzmocnienia obu stron

biorących udział w debacie na temat GPL. Zwolennicy GPL wskazują na krótki żywot większości typów sprzętu komputerowego. Obawiając się ryzyka kupienia przestarzałego produktu, konsumenci biegną do firm, które od lat potrafią się utrzymać na rynku. Z tego powodu rynek oprogramowania stał się areną, na której słabsi są dobijani, a zwycięzcy biorą wszystko⁸. Zwolennicy GPL twierdzą, że rozpowszechnianie oprogramowania z zastrzeżonymi prawami własności prowadzi do monopolu i stagnacji. Silniejsze firmy zużywają całą zapasę życiodajnego tlenu i mniejsze, innowacyjne firmy nie mogą na rynku nawet odetchnąć.

Oponenci GPL argumentują, że jest wprost przeciwnie. Sprzedaż oprogramowania jest tak samo lub nawet bardziej ryzykowna niż jego kupowanie. Bez legalnych gwarancji przypisanych do prywatnych licencji, nie wspominając już o ekonomicznych perspektywach tzw. killer app (tzn. przełomowych technologii pobudzających do życia nowe rynki)⁹, firmy tracą zainteresowanie udziałem w tych przedsięwzięciach. Powoduje to stagnację na rynku i zanikanie innowacji. W przemówieniu wygłoszonym 3 maja w tym samym kampusie Mundie twierdził, że "wirusowa" natura GPL "stanowi zagrożenie" dla wszystkich firm, których konkurencyjność zależy od unikalności ich oprogramowania. Mówił: Podkopuje to fundamenty działalności niezależnego sektora komercyjnego przemysłu programistycznego, gdyż efektywnie uniemożliwia sprzedaż oprogramowania, gdy klienci są zmuszeni do płacenia za produkt, a nie tylko za jego dystrybucję¹⁰.

Znaczny sukces GNU/Linux - systemu operacyjnego będącego połączeniem elementów skupionych wokół jądra chronionego licencją GPL - oraz osiągnięcia Windows w ciągu ostatnich 10 lat wskazują na znaczenie i perspektywy obu typów licencjonowania. Tak czy inaczej walka bieżąca jest wciąż ważnym elementem działalności przemysłu oprogramowania. Nawet najwięksi dostawcy, jak Microsoft, korzystają ze

wsparcia firm trzecich, których narzędzia, aplikacje i gry wspierają i uatrakcyjnają główne platformy systemowe, np. Windows. Opowiadając o gwałtownej ewolucji rynku technologii w ciągu ostatnich 20 lat, nawet bez brania pod uwagę osiągnięć jego własnej firmy, Mundie przestrzegał słuchaczy, by nie dali się uwieść owczemu pędowi do stosowania wolnego oprogramowania:

Dwie dekady doświadczenia wskazują, że model ekonomiczny, w którym własność intelektualna podlega ochronie prawnej, zapewnia zwrot kosztów badań i prac rozwojowych oraz przynosi korzyści ekonomiczne, które stają się udziałem szerokich rzesz odbiorców.

Te ostrzeżenia posłużyły Stallmanowi za punkt wyjścia dzisiejszego przemówienia. Mniej niż miesiąc po ich sformułowaniu rozdrażniony i gotów do rozpoczęcia ataku stanął na podium sali wykładowej.

Dwie ostatnie dekady przyniosły dramatyczne zmiany rynku oprogramowania, a pewnie jeszcze silniej zmieniły samego Stallmana. Zniknął chuderlawy, gładko wygolony haker spędzający całe dni w towarzystwie ukochanego PDP-10. Zamiast niego przed słuchaczami pojawił się mężczyzna w średnim wieku, o przeciętnej tuszy, z długą rabiniczną brodą; człowiek spędzający większość czasu na pisaniu e-maili i odpowiadaniu na nie, zwracający się do kolegów programistów i wygłaszający przemówienia takie jak dzisiejsze. Ubrany w kolorowy T-shirt i poliestrowe brązowe spodnie, wyglądał jak pustynny eremita, który właśnie opuścił siedzibę Armii Zbawienia.

Tłum słuchaczy wyglądał podobnie. Wiele osób przyszło z laptopami i modemami komórkowymi, aby usłyszane słowa zapisywać i przekazywać na użytek społeczności internetowej. Na jedną kobietę przypadało piętnastu mężczyzn, a jedna na siedem lub osiem kobiet przyszła z wypchanym pingwinem, oficjalną linuksową maskotką, lub z wypchanym niedźwiadkiem "teddy bear 11".

Podniecony Stallman opuścił miejsce na mównicy i zasiadłszy w pierwszym rzędzie, zaczął wystukiwać polecenia na klawiaturze przygotowanego laptopa. Robił to przez następnych 10 minut niepomny obecności wzrastającej liczby krążących wokół niego studentów, profesorów i fanów.

Przed rozpoczęciem wykładu musiał a zostać dopełniona akademicka celebra. Ukazanie się Stallmana poprzedziły dwie, zamiast zwyczajowej jednej, zapowiedzi. Pierwszy wystąpił Mike Uretsky, jeden z dyrektorów Stern School's Center for Advanced Technology¹².

[Rysunek 0201]

Richard Stallman około roku 2000. "Zdecydowałem, że stworzę wolny system operacyjny lub polegnę próbując... oczywiście dożywszy słusznego wieku". Zdjęcie dzięki uprzejmości: <http://www.stallman.org/>

Stwierdził, że "zadaniem uniwersytetu jest popieranie debat i podtrzymywanie interesujących dyskusji. To szczególne seminarium spełnia doskonale ten cel. Dyskusja na temat oprogramowania z otwartym kodem źródłowym jest bardzo interesująca."

Zanim przeszedł do następnego zdania, Stallman zerwał się na równe nogi i zamachał rękami jak autostopowicz usiłujący zatrzymać samochód.

"Ja robię wolne oprogramowanie" - rzekł wzbudzając wybuch śmiechu. - "Otwarty kod źródłowy to co innego".

Śmiech przerodził się w aplauz. Sala była wypełniona zwolennikami Stallmana znającymi jego reputację człowieka zwracającego uwagę na ściśle wyśławianie się. Pamiętali też o jego publikacji z roku 1998, w której odzegał się od propozycji zwolenników ruchu otwartego kodu źródłowego. Większość oczekiwała podobnych wybuchów, tak jak fani Jacka Benny'ego

oczekują na jego słynną frazę "Nie wycinaj tego" wygłoszoną w czasie popularnych audycji radiowych.

Uretsky szybko skończył swoją przedmowę, ustępując miejsca Edmundowi Schonbergowi, profesorowi wydziału nauk komputerowych NYU. Jako programista i uczestnik projektu GNU Schonberg wiedział, jak omijać lingwistyczne "miny". Składnie podsumował karierę Stallmana jako współczesnego programisty.

"Richard jest doskonałym przykładem kogoś, kto przeszedł od działania lokalnego do myślenia o globalnym problemie dostępności kodów źródłowych." - mówił Schonberg. "Stworzył zgodną wewnętrznie filozofię, która zmusiła nas do ponownego przemyślenia problemów dotyczących produkcji oprogramowania, znaczenia własności intelektualnej i tego, co obecnie reprezentuje społeczność programistów."

Gorąco powitał Stallmana, który po chwili potrzebnej do zamknięcia laptopa powstał z krzesła i wszedł na podium. Początek tego wystąpienia bardziej przypominał komedię Catskills niż przemówienie polityczne. "Chciałbym podziękować firmie Microsoft za danie mi okazji do dzisiejszego wystąpienia" - rozpoczął od dowcipu Stallman. "Przez kilka ostatnich tygodni czułem się jak autor książki przypadkowo zakazanej".

Dla niewtajemniczonych wygłosił krótką, choć konkretną, gorącą pochwałę wolnego oprogramowania. Porównał program komputerowy do kucharskiej recepty. W obu przypadkach mamy do czynienia z dokładnym przepisem, który krok po kroku prowadzi nas aż do wykonania całego zadania, ale pozwala na dokonanie pożądaných modyfikacji - jeżeli mamy na to ochotę lub okoliczności tego wymagają. "Nikt was nie zmusza do dokładnego przestrzegania receptury. Możecie coś opuścić lub coś dodać. Jeżeli lubicie grzyby, możecie włożyć więcej grzybów, a jeżeli nie lubicie soli lub lekarz zabronił wam jej używać, możecie dać jej mniej niż w przepisie."

Twierdził, że najważniejsze, by programy komputerowe, tak jak

przepisy kucharskie, był y ogólnodostępne. Dając gościowi przepis, kucharz traci niewiele ponad czas i papier potrzebny do jego zapisania. W przypadku programów komputerowych wysiłek jest jeszcze mniejszy; zwykle wystarcza kilka kliknięć myszą i nieco elektryczności. W obu przypadkach osoba dająca informacje korzysta podwójnie: wzmacnia przyjaźń i zwiększa prawdopodobieństwo uzyskania w rewanżu innych przepisów/programów.

"Wyobraźcie sobie, co by się działo, gdyby przepisy kucharskie zapakować w szczelne czarne pudełka" - pytał Stallman, podnosząc głos. - "Nie wiedzielibyście, jakie składniki zostały użyte, więc nie wiedzielibyście również, co możecie zmienić. A gdybyście mimo to jakichś zmian dokonali albo sporządzili kopię dla przyjaciela, bylibyście ścigani jako piraci, a ścigający staraliby się osadzić was w więzieniu na długie lata. Taki świat może się wydawać straszny dla tych, którzy przywykli do dzielenia się z innymi posiadanymi informacjami. Ale właśnie taki jest świat własności prywatnej oprogramowania. Jest to świat, w którym zwykła ludzka przyzwoitość została zakazana". Dla uzasadnienia swojego zdania opowiedział epizod z drukarką laserową Xeroxa. Historia ta była chwytem retorycznym równie użytecznym jak porównanie z przepisem kucharskim. Pełniąc funkcję przypowieści, uświadamiała, jak dramatycznie szybkie zmiany następują w świecie oprogramowania. Cofnął słuchaczy myślą do czasów sprzed "jednokliknięciowego" dokonywania zakupów w Amazon.com, czasów bez Windows i bez baz danych Oracle' a gdy znaki firmowe nie przypominały o prawach własności oprogramowania.

Opowiedział to wszystko z wprawą i praktyką oskarżyciela publicznego dostarczającego ostatecznych, przygważdżających przeciwnika argumentów. Gdy doszedł do historii z drukarką laserową i odmawiającym mu udostępnienia kodu źródłowego profesorem Carnegie Mellon, zrobił efektowną pauzę.

"On nas zdradził" - powiedział, - "ale krzywdę uczynił nie tylko nam. Prawdopodobnie uczynił ją każdemu z was".

Mówiąc "was", skierował palec wskazujący w stronę jednego z nic niepodjejrzejawających słuchaczy. Brwi wskazanego uniosły się w zdziwieniu, ale wzrok Stallmana już przesunął się dalej. Powoli i z rozważą wskazał kogoś innego, chichoczącego nerwowo: "I sądzę, że skrzywdził również ciebie" - rzekł do jegomości w trzecim rzędzie.

Wskazany przestał chichotać, co widownia przyjął aśmiechem. Gest wyglądał scenicznie i taki był wrzeczywistości. Zawsze, gdy dochodził do końca historii o drukarce laserowej, czynił to z doskonałością showmana. "Wyrządził zło prawdopodobnie każdemu z obecnych tu na sali, za wyjątkiem tych niewielu, którzy w roku 1980 jeszcze się nie narodzili" - mówił dalej, wzbudzając gromki śmiech. "Twierdzę tak, gdyż odmówił współpracy z całą ludzkością, ze wszystkimi mieszkańcami planety Ziemia".

Począkał aż komentarz dotrze do świadomości słuchaczy i dodał "Zdradził was, podpisując umowę o nieujawnianiu informacji". Przekształcił cenie Richarda Matthew Stallmana ze sfrustrowanego akademika w politycznego przywódcę przemawiającego już od 20 lat świadczy o jego upartej naturze i wyjątkowej woli oraz jasno określonej wizji i wartościach, które głosi, wytrwale budując ruch wolnego oprogramowania. Dodać do tego należy autorstwo programów najwyższej jakości, które ustalił yjego legendarną reputację programisty. Wielu zwolenników uważa jednak ruch wolnego oprogramowania za jego największe osiągnięcie.

Co najważniejsze, Stallman działa w świecie o zmiennej naturze potęg politycznych, który jednak coraz bardziej uzależnia się od komputerów i sterujących ich działaniem programów.

Być może dlatego w czasach, gdy wiele osobistości związanych z najnowszą technologią odchodzi w zapomnienie, jego gwiazda błyszczy coraz silniejszym światłem. Od czasu rozpoczęcia projektu GNU w roku 1984¹⁴, Stallman był kolejno ignorowany, wyśmiewany, oczerniany i atakowany wewnątrz ruchu wolnego oprogramowania i poza nim. Pomimo to projekt

GNU pokonuje na swej drodze kolejne kamienie milowe, aczkolwiek czasem z pewnym opóźnieniem wynikającym z coraz większych wymagań rynku, o wiele bardziej złózonego niż w chwili rozpoczęcia działania przed 18 laty. To samo można powiedzieć o ideologii GNU drobiazgowo dopracowywanej przez samego Stallmana.

Aby zrozumieć ukryte mechanizmy tego działania, należy dokładnie poznać wypowiedzi zarówno samego Stallmana, jak i ludzi z nim współpracujących oraz tych, którzy go zwalczają. Jego charakter nie jest skomplikowany i pasuje do starego powiedzenia: "koń jaki jest, każdy widzi".

Eden Moglen, doradca prawny Fundacji Wolnego Oprogramowania i profesor prawa na w Columbia University, twierdzi, że "aby poznać ludzką naturę Stallmana, wystarczy się przyjrzeć poszczególnym jej częściom i złożyć wszystko w całość. Ekscentryczność, którą niektórzy uważają za barierę utrudniającą poznanie go, jest jego właściwą naturą. Prawdziwe są jego silne poczucie frustracji i nadawanie wielkiego znaczenia przestrzeganiu zasad etycznych oraz całkowita niezdolność do kompromisu. Jest to podstawa jego działań i fundament tego, co zbudował".

Wytył umaczenie drogi, która doprowadził go od epizodu z drukarką laserową do walki z najbogatszą korporacją świata, nie jest łatwym zadaniem. Wymaga szczegółowego przemyślenia powodów, dla których sprawa własności oprogramowania stała się tak istotna dla współczesnego społeczeństwa. Wymaga także przemyślanej oceny człowieka, który jak wielu przywódców politycznych przed nim zrozumiał podatność ludzkiej świadomości. Wymaga to umiejętności interpretowania mitów i politycznych obciążeń słów kojarzonych z jego osobą. Na koniec, wymaga to zrozumienia geniuszu programistycznego Stallmana oraz wzlotów i upadków na drodze wykorzystania tego geniuszu w innych dziedzinach.

Poproszony o podsumowanie swojej drogi, Stallman zgadza się, że istotą jest spostrzeżona przez Moglena fuzja osobowości i

zasad. "Upór to moja mocna strona" - przyznaje. "Wielu ludzi biorących się za rzeczy trudne zniechęca się i rezygnuje w wyniku piętrzących się trudności. Ja nigdy nie rezygnuję". Wierzy także w ślepy traf. Mógł nigdy nie mieć kłopotów z drukarką Xeroxa, mógł nie doznać osobistych i politycznych problemów, które zakończył yjego karierę pracownika MIT, mogło nie mieć miejsca pół tuzina innych zdarzeń, które stały y się istotnymi czynnikami w jego życiu. Z ł atwością wyobraża sobie swoje życie przebiegające zgoła odmiennie. Jak już powiedzieliśmy, Stallman jest wdzięczny siłom i okolicznościom, które odmienił yjego los.

"Po prostu mam odpowiednie uzdolnienia i skł onności" - mówi do słuchaczy, podsumowując rozpoczęcie projektu GNU. "Nikt oprócz mnie się tym mnie zajął , więc zapewne był ęm przeznaczony do tej roli. Muszę podołać zadaniu. Jeżeli nie ja, to kto?"

Przypisy

1 Ufundowany w roku 1831 Uniwersytet Nowojorski jest największą prywatną uczelnią w Stanach Zjednoczonych. Skupia ponad 40 000 studentów. Jego liczne instytucje są rozlokowane głównie w pięciu centrach na Manhattanie. Dokładniejsze informacje można znaleźć pod adresem: <http://www.nyu.edu/>

- przyp. tł um.

2 Free Software Foundation (FSF) - dokładniejsze informacje po polsku można znaleźć pod adresem:

<http://www.gnu.org/home.pl.html> - przyp. tł um.

3 Nagroda Fundacji MacArthura - przyznawana corocznie około 20 - 40 obywatelom lub rezydentom USA za "wyjątkowe zasługi i obietnicę kontynuacji i rozszerzenia twórczej pracy". Bywa nazywana "nagrodą geniuszy". W roku 2002 wynosił a500 tys. dolarów płatnych w czterech ratach w ciągu pięciu lat - przyp. tł um.

4 Nagroda w wysokości 5000 dolarów przyznawana corocznie wybitnym młodym komputerowcom w wieku poniżej 35 lat. Listę nagrodzonych można znaleźć pod adresem: http://www.acm.org/awards/awards_hopper.html.

Grace Murray Hopper (1906-1992), którą upamiętnia nagroda, w roku 1952 stworzył akompilator A-O, pierwszy program, tłumaczący symbole języka wysokiego poziomu na kod maszynowy. Więcej informacji o niej oraz o innych kobietach pionierkach nauk komputerowych można znaleźć pod adresem: <http://www.cs.yale.edu/homes/tap/past-women-cs.html> - przyp. tłum.

5 Prestiżowa nagroda w wysokości 100 mln jenów przyznawana przez Fundację Takeda za wybitne osiągnięcia intelektualne i inżynierskie służące ludzkości. Dodatkowe informacje o nagrodzie można znaleźć w serwisie WWW fundacji: <http://www.takeda-foundation.jp/en/award/takeda/> - przyp. tłum.

6 GNU General Public Licence (w skrócie GNU GPL lub tylko GPL) - patrz http://pl.wikipedia.org/wiki/Powszechna_Licencja_Publiczna_GNU - przyp. tłum.

7 Obecnie GPL stracił nieco swej mocy. Zgodnie z rozdziałem 10 Powszechnej Licencji Publicznej GNU, wersja 2 (1991), wirusowa natura licencji zależy w dużym stopniu od woli Fundacji Wolnego Oprogramowania, która uznaje, czy dany program należy uznać za pochodny i czy należy zastąpić bieżącą licencję licencją GNU.

Jeśli chcesz włączyć części Programu do innych wolnych programów, których warunki rozpowszechniania są inne, zwróć się pisemnie do autora z prośbą o pozwolenie. W przypadku oprogramowania objętego przez Fundację prawem autorskim, napisz do Fundacji; czasami czynimy od tego odstępstwa. W naszej decyzji kierujemy się dwoma celami: utrzymania wolnego statusu wszystkich pochodnych naszego wolnego oprogramowania oraz - generalnie - promowania współudziału i

wielokrotnego stosowania oprogramowania (tł umaczenie fragmentu licencji podane na stronie Fundacji Wolnego Oprogramowania: <http://gnu.org.pl/text/licencja-gnu.html>).

"Porównanie czegoś do wirusa jest bardzo przykre" - twierdzi Stallman

- "Lepiej porównać GPL do rośliny pająka (spider plant - Chlorophytum), która tam się rozrasta, gdzie jest aktywnie przycinana".

Więcej informacji na temat Powszechnej Licencji Publicznej GNU można znaleźć pod adresem:

<http://www.gnu.org/copyleft/gpl.html>.

8 Patrz: Shubba Ghosh "Revealing the Microsoft Windows Source Code" Gigalaw.com (styczeń 2000),

<http://www.gigalaw.com/articles/ghosh-2001-01-p1.html> (obecnie strona jest niedostępna z powodu reorganizacji serwisu Gigalaw.com - w nawiasie przyp. tł um).

9 Killer app nie muszą być aplikacjami z prywatnym ograniczonym prawem własności, czego dowodem jest legendarna przeglądarka Mosaic - program, którego licencja zezwalał az pewnymi ograniczeniami na niekomercyjne przekształcanie i tworzenie programów pochodnych. Myślę, że Czytelnicy pojęli już podstawową zasadę, że rynek oprogramowania to wielka loteria. Im większe spodziewane dochody, tym więcej chętnych do uczestnictwa. Dobrym podsumowaniem zagadnień dotyczących "killer app" jest artykuł : Philip Ben-David: Whatever Happened to the "Killer App?" e-Commerce News (7 grudnia 2000) -

<http://www.ecommercetimes.com/perl/story/5893.html>

10 Patrz: Craig Mundie: The Commercial Software Model - fragmenty wykładu wiceprezesa Microsoft Corporation ogłoszonego 3 maja 2001 w New York University Stern School of Business: <http://www.microsoft.com/presspass/features/2001/may01/05-03csm.asp>

11 Teddy Bear - przezwisko, które przyłgnął odo prezydenta Teodora Roosevelta po opublikowaniu jego karykatury w roku

1906.

Miś teddy bear (pisany czasem z wielkiej, a czasem z małej litery) stał się maskotką i symbolem jednej z internetowych społeczności.

Więcej informacji można znaleźć pod adresem:

<http://www.tbonnet.com/>

- przyp. tł um.

12 Centrum Zaawansowanych Technologii Szkół ySterna -
przyp. tł um.

13 Catskills comedy - mniej więcej od stulecia nowojorczyacy żydowskiego pochodzenia, żądni odpoczynku w górach, dobrego jedzenia i przebywania we własnym środowisku kulturowym, zaczęli budowę letnich domów, hoteli i pensjonatów w rejonie Catskills. Zjawisko to nasiliło się w pierwszej połowie XX wieku i spowodowało powstanie okręgu o unikalnej w skali światowej atmosferze i kulturze, na podobieństwo niepowtarzalnej kultury przedwojennego żydowskiego krakowskiego Kazimierza oraz wielu miast i miasteczek wschodniej Rzeczypospolitej. Catskills comedy to komedia w odrębnym stylu Catskills, jak odrębny był styl dawnych żydowskich szmoncesów. Nieco informacji na ten temat można znaleźć pod adresem:

http://www.brown.edu/Administration/News_Bureau/1995-96/95-062i.html

- przyp. tł um.

14 Akronim GNU oznacza "GNU is not Unix". We fragmencie wystąpienia 29 maja 2001 w New York University Stallman tak wyjaśnia jego pochodzenie:

Hakerzy zawsze poszukują zabawnych i niezwykłych nazw dla swoich programów. Znalezienie dobrej nazwy to połowa zabawy piszącego program. Istnieje tradycja stosowania rekursywnych akronimów, sugerujących, że tworzony program jest podobny do już istniejącego... Szukałem takiego akronimu dla "Coś nie jest Uniksem". Wypróbowałem wszystkie 26 liter, by stwierdzić, że nie da się z nich złożyć żadnego odpowiedniego słowa, więc

zdecydowałem się na sprzeczność z językiem. Tylko w ten sposób mogłem otrzymać trzyliterowy akronim oznaczający, że "Coś nie jest Uniksem". Wypróbowując różne ustawienia liter, doszedłem do "GNU is not Unix". To było to.

Choć jest miłośnikiem kalamburów, zaleca wyraźne wymawianie "g" na początku słowa GNU, aby uniknąć pomylenia z nazwą antylopy afrykańskiej "gnu" (*Connochaetes gnou*) oraz ze słowem "new" (nowy, nowe itp.). "Pracujemy nad tym już od 17 lat, więc to nic nowego" - powiada.

Źródło: notatki autora i internetowy zapis wystąpienia Stallmana w NYU 29 maja 2001:

<http://www.gnu.org/events/rms-nyu-2001-transcript.txt>.

3. PORTRET HAKERA W WIEKU MŁODZIEŃCZYM

Matka Richarda Stallmana, Alice Lippman, pamięta moment, w którym uświadomiła sobie, że jej syn posiada szczególny dar. "Myślę, że miał wtedy osiem lat" - wspomina.

Był rok 1961 i Lippman, świeżo po rozwodzie, chciał a spędzić weekend z rodziną w niewielkim jednopokojowym mieszkaniu na manhattańskim Upper West Side. Kartkując *Scientific American*, dotarł a do ulubionego działu, prowadzonej przez Martina Gardniera kolumny autorskiej pod tytułem "Gry matematyczne". Jako pracująca na zastępstwie nauczycielka sztuki lubiła kolumnę Gardniera, gdyż drażniąc trudnymi pytaniami, zmuszała do myślenia. Richard z nosem ukrytym w książce siedział na sąsiedniej sofie, więc postanowił a wziąć się za rozwiązywanie cotygodniowej łamigłówki.

"Nie byłem w tym najlepsza" - przyznaje, - "ale jako artystka, uważałam, że pomaga mi to w przełamywaniu barier myślowych".

Przypomina sobie, że wówczas nie wiedział a, jak "ugryźć" łamigłówkę, i gdy już miała zamiar z niesmakiem odrzucić

pismo, poczuła delikatne pociągnięcie za rękaw bluzki.

"To był Richard" - wspomina, - który chciał się dowiedzieć, czy nie potrzebuje pomocy".

Spoglądając na przemian na łamigłówkę i syna, na początku podeszła sceptycznie do propozycji. "Spytałam go, czy już przeczytał pismo. Potwierdził i dodał, że rozwiązał łamigłówkę, po czym zaczął mi wyjaśniać, jak to zrobił".

W miarę słuchania wyjaśnień syna jej sceptycyzm szybko przeradzał się w niedowierzanie. "Zawsze wiedziałam, że z niego zdolne dziecko, ale po raz pierwszy uświadomiłam sobie, jak bardzo jest rozwinięty".

Trzydzieści lat później kwituje to wspomnienie uśmiechem.

"Prawdę mówiąc, nie sędzę, abym kiedykolwiek pojęła, jak należało rozwiązać tamtą łamigłówkę. Pamiętam jedynie swoje zdumienie, że on to potrafi".

Siedząc w jadalni trzypokojowego mieszkania, tego samego, do którego przeniósł się wraz z synem w roku 1967, po ślubie z nieżyjącym już drugim mężem Mauricem Lippmannem, promienieje mieszaniną dumy żydowskiej matki i radości, jaką jej sprawia wspomnianie dziecięcych lat syna. Na stojącej opodal komodzie widnieje zdjęcie o wymiarach 8 na 10 cali przedstawiające Stallmana w dostojeństwie długiej brody i togi doktora honoris causa. W jego cieniu pozostają zdjęcia bratanicy i siostrzeńców, ale nim gość zdoła się przyjrzeć, Alice Lippman rozluźnia nastrój dowcipną uwagą.

"Po otrzymaniu doktoratu University of Glasgow Richard upierał się, abym wziął to zdjęcie, mówiąc: 'Wiesz, mam, to mój pierwszy stopień naukowy'".

Takie komentarze ukazują rodzaj humoru, który towarzyszył wzrastaniu cudownego dziecka. Na każdą wzmiankę o uporze i niezwykłym zachowaniu syna Alice potrafi opowiedzieć tuzin anegdot.

"Był bardzo konserwatywny" - opowiada, zacierając ręce żartobliwym gestem. "Najostrzejsze dyskusje odbywały się przy tym stole. Należałam do pierwszej grupy nauczycieli szkół

publicznych, którzy założyli związek zawodowy, za co Richard bardzo się na mnie gniewał. Uważał, że związki zawodowe powodują korupcję. Był także przeciwnikiem ubezpieczeń społecznych. Uważał, że ludzie mogą osiągnąć znacznie więcej, samodzielnie inwestując pieniądze. Kto się mógł spodziewać, że po 10 latach wyrośnie z niego idealista. Pamiętam, że jego przyrodnia siostra mówiła: "Co z niego będzie? Wyrośnie na faszystę".

Alice Lippman wyszła za Daniela Stallmana w roku 1948. Rozesła się z nim w 1958 i przez niemal 10 lat samotnie wychowując Richarda, wpoiła mu awersję do autorytetów oraz żądę wiedzy. Był to czas ścierania się dwóch sił, ona i syn staczali ciągłe boje.

"Nigdy nie chciał jeść" - wspomina. "Trwał o to od skończenia 8 lat do końca szkoły średniej w roku 1970. Gdy wołał mnie na obiad, nigdy nie słyszał. Zanim to do niego dotarło, musiał mnie krzyczeć dziewięć lub dziesięć razy. Całkowicie zatapiał się w myślach".

Stallman pamięta to podobnie, ale z pewnym podtekstem.

"Lubiłem czytać" - wspomina. - "Jeżeli miałem ochotę poczytać, a matka wołała mnie do kuchni albo kazała iść spać, nie miałem ochoty słuchać. Nie rozumiałem, dlaczego mam przerywać czytanie. Nie było żadnego powodu, dla którego miałam prawo mi nakazywać, co mam robić. Podstawowe idee demokracji, o których czytałam, stosowałem do siebie samego. Nie rozumiałem, dlaczego nie miałoby dotyczyć dzieci".

Wiarę w osobistą wolność wyłamywania się spod władzy autorytetów rozciągał na szkołę. Wyprzedzając kolegów klasowych o dwa lata, już w wieku 11 lat doświadczył frustracji, które zwykle są udziałem utalentowanego ucznia szkoły publicznej. Niedługo po rozwiązaniu przez Richarda prasowej kampanii jego matka przeżyła pierwszą z wielu wywiadówek.

"W ogóle odmawiał pisania wypracowań" - wspomina dawne problemy. "Myślę, że ostatnie, jakie napisał przed klasą

maturalną, był oesejem na temat historii systemów liczbowych przygotowanym na polecenie nauczyciela w czwartej klasie". Uzdolniony we wszystkich dziedzinach, które wymagały myślenia analitycznego, ciążył ku matematyce i naukom ścisłym, zaniedbując wszystkie pozostałe. To, co inni nauczyciele traktowali jako jednostronne ograniczenie, matka uważała za głód wiedzy. Matematyka i nauki ścisłe oferowały tak wiele do poznania w porównaniu z przedmiotami, do których jej syn miał mniej zdolności. W wieku 10 lub 11 lat, gdy inni chłopcy z jego klasy rozpoczęli naukę gry w futbol, jej syn - jak wspomina - zaczął wracać do domu sfrustrowany. "Nie chciał grać tak fatalnie, ale po prostu miał kłopoty z koordynacją ruchów. To go wprawiało we wściekłość". Złość spowodowała jeszcze silniejsze skupienie się na matematyce i naukach ścisłych. Jego pasja nawet przy rzeczywistej pracy naukowej mogła sprawiać kłopoty. Pochłaniając książki matematyczne w wieku siedmiu lat, nie odczuwał potrzeby wdawania się w rozmowy z dorosłymi. W latach nauki w middle-school² matka wpadła na pomysł, by wynająć studenta z sąsiedniego Columbia University, który mógłby zajmować się Richardem jak starszy brat. Po pierwszym spotkaniu student wyszedł i nigdy nie powrócił. "Wyobrażam sobie, co Richard musiał mu nagadać" - spekuluje matka. Kolejna matczyzna anegdota dotyczy lat sześćdziesiątych, czasu tuż po zdarzeniu z łamigłówką. Mniej więcej w wieku lat siedmiu, w dwa lata po rozwodzie matki, Richard zajął się puszczaniem modeli rakiet w sąsiednim Riverside Drive Park. To, co zaczęło się jako niewinna zabawa, rychło stało o się poważnym zadaniem badawczym połączonym z zapisywaniem danych o każdym starcie. Podobnie jak zainteresowanie grami matematycznymi, tak i ta pasja syna nie wzbudzała jej zbytniego zainteresowania. Było tak do czasu ważnego wystrzelenia rakiety przez NASA. Spytał wówczas Richarda, czy zamierza to oglądać w telewizji. "Wściekł się i jedyne, co mogłam zrozumieć, to to, że jeszcze

czegoś nie opublikował" - wspomina. - "Naprawdę uważał, że ma coś istotnego do zakomunikowania NASA".

Takie anegdoty świadczą o jego wczesnym silnym angażowaniu się w wykonywane zadania. Będzie go to cechował przez całe życie. Gdy inne dzieci bawiły się, on siedział w pokoju i czytał. Gdy inne dzieci naśladowały Johnny'ego Unitasa³, on naśladował Wernera von Brauna. "Był ęm dziwakiem" - zwięźle podsumował swoje młodsze lata w wywiadzie z 1999 r. - "W pewnym wieku nie miał ęm żadnych przyjaciół poza nauczycielami⁴".

Choć oznaczało to dla niej częstsze starcia ze szkołą, Alice Lippman zdecydowała się pozwalać synowi na realizowanie jego pasji. W wieku 12 lat Richard uczestniczył w letnich obozach naukowych, a w ciągu roku szkolnego - w dodatkowych zajęciach w szkole prywatnej. Nauczyciel poradził jej zapisanie syna do uczestnictwa w Columbia Science Honors Program⁵, który w erze posputnikowej uruchomiono dla uzdolnionej młodzieży z nowojorskich szkół średnich i college'ów. W ten sposób Stallman zaczął sobotnie wędrowki do campusu Columbia University. Jak wspomina Dan Chess, kolega z wykładów w Columbia Science Honors Program, Stallman wydawał się być "zakrecony" nawet w tej grupie miłośników matematyki i nauk ścisłych. "Wszyscy byliśmy dziwaczni, nieprzystosowanymi do życia młodzieżą, zakochanymi w matematyce i naukach ścisłych, ale Richard wyróżniał się nawet wśród nas" - wspomina Chess, obecnie profesor matematyki w Hunter College. "Był bystry jak diabeł. Widział ęm wielu bystrych ludzi, ale on był zewszystkich najbystrzejszy". Inny kolega z Columbia Science Honors Program, Seth Breidbart, potwierdza tę opinię. Programista, którego ze Stallmanem łączyły wspólne zainteresowania science-fiction i konwencjami tego gatunku, wspomina go jako krótko wystrzyżonego "straszego" piętnastolatka.

"Ciężko to opisać" - mówi Breidbart. - "Nie był niedostępny. Był strasznie spięty. O dużej wiedzy, ale pod pewnymi względami

niereformowalny".

Takie opinie rodzą spekulacje: czy określenia "spięty" i "niereformowalny" to cechy, które możemy dzisiaj uznać za zwykłe dziecięce wyłamywanie się z reguł środowiska? Pismo Wired z grudnia 2001 r. w artykule "The Geek Syndrome"⁶ kreśli portrety kilku naukowo uzdolnionych dzieci, u których zdiagnozowano zespół Aspergera⁸ z cechami silnego autyzmu. Pod wieloma względami przytoczone przez Wired wspomnienia rodziców tych dzieci niejasno kojarzą się z tym, co opowiada matka Stallmana. Nawet on sam czasem pozwala sobie na psychiatryczne konotacje. W roku 2000, kreśląc swą sylwetkę dla Toronto Star, opisał siebie jako "będącego na granicy autyzmu"⁹, co odpowiada jego wykazywanej przez całe życie chęci społecznego i emocjonalnego izolowania się od otoczenia i równie długi utrwałym usiłowaniom przezwyciężenia tych skłonności.

Takie spekulacje wynikają, oczywiście, z częstych ostatnio rozważań na temat swobodnego i nieodpowiedzialnego zachowania związanego z tzw. behavioral disorder¹⁰. Jak twierdzi Steve Silberman, autor The Geek Syndrome, amerykańscy psychiatrzy używają ostatnio nazwy zespołu Aspergera jako wygodnego parasola kryjącego wiele różnorodnych cech behawioralnych: od ograniczonych zdolności motorycznych i niemożności współżycia z otoczeniem po wybitną inteligencję połączoną ze szczególnym zainteresowaniem liczbami, komputerami i systemami¹¹. Biorąc pod uwagę tak szeroki zasięg "parasola" zwanego zespołem Aspergera, Stallman może przypuszczać, że gdyby urodził się 40 lat później, w ten sposób zostałby zdiagnozowany. Zapewne spotkałby to wielu jego kolegów-komputerowców.

"Możliwe, że miałem coś takiego" - stwierdza - "ale z drugiej strony jedną z cech charakterystycznych tego zespołu są trudności w zachowaniu rytmu. Ja umiem tańczyć, a nawet sprawia mi przyjemność poruszanie się w takt melodii o

najbardziej skomplikowanym rytmie. Ale to wszystko nie jest do końca wyjaśnione".

Chess wyklucza tego rodzaju diagnozę: "Nigdy nie przypuszczałem, że cierpi na coś takiego. Był po prostu nietowarzyski, ale wszyscy byliśmy tacy".

Matka Richarda uznaje tę interpretację za zabawną. Jednakże wspomina kilka historii z jego dzieciństwa, które mogą stanowić pożywkę dla dalszych spekulacji. Dwie spośród opowiadanych przez nią anegdot pasują do ważnej cechy autyzmu, jaką jest nadwrażliwość na dźwięki i kolory. "Gdy Richard był dzieckiem, zabraliśmy go na plażę" - powiada matka. - "Zaczął marudzić o dwie lub trzy przecznice wcześniej, nim dotarliśmy do wody. Dopiero gdy to się powtórzył o, za trzecim razem zrozumieliśmy, jaka była przyczyna jego narzekania. Szum fal ranił jego uszy". Pamięta również podobną reakcję na kolory. "Moja matka miała a intensywnie rude włosy i zawsze, gdy się pochylała nad Richardem, aby go wziąć na ręce, uderzał w płacz".

Ostatnio zaczął czytać książki o autyzmie i sądzi, że zdarzenia te były czymś więcej niż zwykłym zbiegiem okoliczności. "Sądzę, że Richard miał pewne skłoności autystyczne. Szkoda, że wówczas tak niewiele o tym wiedziano".

Jednakże uważa, że z czasem jej syn przystosował się. W wieku siedmiu lat lubił stać w pobliżu przedniej szyby wagonu metra, odczytywać, zapamiętywać i porządkować w głowie podziemny labirynt komunikacyjny. Było to jego hobby połączone z przystosowywaniem się do słuchania dźwięków towarzyszących ruchowi wagonów. "W pierwszej chwili hałas wywoływał w nim odrazę, po czym jego nerwy uspokajały się". Jak wspomina, jej syn prawie zawsze wykazywał podniecenie, energię i umiejętność kontaktowania się z innymi, taką samą jak wszyscy normalni chłopcy. Dopiero po serii traumatycznych zdarzeń, które spadły na ich rodzinę, zamknął się w sobie i stał się introwertykiem.

Pierwszym zdarzeniem traumatycznym był jej rozwód z

Danielem Stallmanem, ojcem Richarda. Choć oboje starali się przygotować chłopca na ten cios, spowodował on spustoszenia w jego psychice. "Gdy po raz pierwszy powiedzieliśmy mu o tym, co ma nastąpić, zdawał się nie przywiązywać do tego wagi" - wspomina. "Rzeczywistym uderzeniem była dla niego moja i jego przeprowadzka do nowego mieszkania. Pierwszym jego pytaniem było 'Gdzie meble tatusia?'".

W ciągu następnych dziesięciu lat pięć dni w tygodniu spędzał w mieszkaniu matki na Manhattanie, a weekendy - w domu ojca w Queens. Te odejścia i powroty pozwoliły mu na ocenę różnego stylu życia obojga rodziców i wywołały trwały uraz przejawiający się w niechęci do posiadania i wychowywania dzieci. Gdy mówi o ojcu, weteranie II wojny światowej, który zmarł w roku 2001, wyraża na przemian złość i szacunek. Z jednej strony pamięta go jako człowieka, który z moralnego obowiązku, aby być bardziej użytecznym dla aliantów, nauczył się francuskiego, a z drugiej - jako ojca, który zawsze wiedział, w jaki sposób sformułować odmowę, aby cios był najokrutniejszy¹².

"Mój ojciec miał straszny charakter" - twierdzi Richard. - "Nigdy nie narzekał, ale zawsze znalazł sposób, aby krytykować spokojnie, na zimno i miarowo."

Na temat mieszkania z matką wyraża się mniej dwuznacznie.

"To był awgna" - powiada. - "W swym nieszczęściu, mówiąc 'Idę do domu', mówił mi o czymś nieistniejącym, o czymś, czego nigdy nie miałem'".

W ciągu kilku lat po rozwodzie rodziców ażylem stał się dla niego dom dziadków ze strony ojca. Gdy miał około dziesięciu lat, dziadkowie zmarli - jedno wkrótce po drugim. Był to dla niego wielki cios. "Uciekałem tam, do miejsca, gdzie znajdowałem spokój i gdzie mnie kochano" - wspomina. - "Przed pójściem do college'u był to jedyne moje miejsce na świecie".

Matka uważa śmierć dziadków za drugie traumatyczne zdarzenie, które spotkało syna. "Bardzo się tym przejął" -

twierdzi. "Był do nich bardzo przywiązany. Do czasu ich śmierci był bardzo otwarty, przewodził paczce rówieśników. Potem zamknął się w sobie, stał się wyobcowany".

Z perspektywy samego Stallmana emocjonalna izolacja była jedynie próbą poradzenia sobie z końcem okresu dojrzewania. Nazywa swoje młode lata "czystym horrorem" i twierdzi, że często czuł się jak głuchy wśród tłumy dyskutującego o muzyce.

"Często wydawało mi się, że nie jestem w stanie pojąć, co inni mówią" - wspomina, uważając, że emocjonalna zasłona odgradziła go od świata rówieśników i dorosłych. "Pojmowałem słowa, ale coś tkwiącego głębiej nie pozwalało mi złożyć ich w zrozumiałą całość. Nie mogłem pojąć, dlaczego jedni są zainteresowani tym, co mówią inni."

Kończący się okres dojrzewania wzmocnił jego poczucie indywidualizmu. Gdy większość jego rówieśników zapuszczała wówczas długie włosy, on strzygł się krótko. Gdy całe pokolenie słuchało rocka, on wolał muzykę klasyczną. Zawołany miłośnik science-fiction, magazynu "Mad" i późnonocnych programów telewizyjnych, kultywował swoją odrębność hodowaną na gruncie niezrozumienia rodziców i rówieśników.

"A te kalambury" - matka Richarda wspomina z przerażeniem okres, gdy był nastolatkiem - "przy stole nie można było słowa powiedzieć, bo wszystko przekręcał w jakiś kalambur zwrócony przeciw mówiącemu".

Zapamiętywał dowcipy i zabawy słowne, by poza domem opowiadać je dorosłym, którzy byli skłonni pobyliwać u talentowanego dzieciaka. Pierwszym był instruktor na obozie letnim, który wręczył dwunastolatkowi drukowaną instrukcję obsługi komputera IBM 7094. Dla chłopaka zafascynowanego liczbami i nauką był to dar niebios¹⁴. Przed końcem lata potrafił już pisać na papierze programy zgodne ze specyfikacjami wewnętrznymi 7094, zakładając, że znajdzie okazję do ich uruchomienia.

Pierwszy komputer osobisty miał się pojawić dopiero po

dekadzie, więc wszystko wskazywało na to, że przyjdzie mu długo czekać na ziszczenie pragnień. Okazja trafił a się, gdy był w pierwszej klasie junior high school. Po rozdaniu świadectw zaczął pracę w obecnie już nieistniejącym IBM New York Scientific Center w centrum Manhattanu. Spędził tam lato, pisząc swój pierwszy program: preprocesor dla 7094 w języku PL/I. "Najpierw pisał em w PL/I, by potem przejść na assembler, gdy okazało się, że program w PL/I jest za duży do uruchomienia na tym komputerze" - wspomina.

Po opuszczeniu IBM Scientific Center zatrudnił się jako asystent laboratoryjny na wydziale biologii Rockefeller University. Choć jego zainteresowania już wyraźnie zaczęły się skupiać na matematyce i fizyce, jego analityczny sposób myślenia wywarł wrażenie na dyrektorzem laboratorium, który w kilka lat później niespodzianie zadzwonił do matki Richarda. "Chciał się dowiedzieć, co Richard robi. Był zdumiony, że zajmuje się komputerami, bowiem przypuszczał, że wyrośnie z niego doskonały ybiolog".

Jego zdolności analitycznego myślenia wzbudzały podziw także na zajęciach Columbia Science Honors Program, nawet wówczas gdy zachowywał się irytująco. "Zwykle raz lub dwa w ciągu godziny znajdował jakiś błąd w wykładzie" - wspomina Breidbart "i nie brakowało mu śmiałości do natychmiastowego wytknięcia go wykładowcy. Dzięki temu cieszył się wielkim respektem, ale znacznie mniejszą popularnością".

Przypomnienie przytoczonej przez Breidbarta anegdoty wywołało krzywy uśmiech na twarzy Stallmana. "Może czasem się ich czepiałem" - twierdzi, - "ale właśnie wśród nauczycieli znajdowałem bratnie dusze, bo w przeciwieństwie do uczniów chcieli się uczyć. Szczeniaki w większości nie miały na to ochoty, przynajmniej nie w tym samym stopniu".

Sobotnie przebywanie wśród uzdolnionej młodzieży niewątpliwie zwiększyło jego pewność siebie i chęć kontaktowania się z rówieśnikami. W miarę zbliżania się czasu studiów jego zainteresowania, tak samo jak większości kolegów

z Columbia Science Honors Program, stawał się coraz bardziej ukierunkowane, co powodowało zawężanie możliwości wyboru uczelni. W końcu pozostały tylko dwie: Harvard i MIT. Słyszając, że syn wybiera się na uczelnię należącą do Ivy League¹⁵, Alice Lippman poważnie zainteresowała się tą sprawą. Jako piętnastoletni uczeń junior high school Stallman wciąż miewał starcia z nauczycielami i administracją szkoły, wciąż z nimi polemizował. W poprzednim roku otrzymał ocenę A z historii Ameryki, chemii, francuskiego i algebry, ale te osiągnięcia przyćmiewało F16 z angielskiego, co było wynikiem bojkotowania przez niego wszelkich wypracowań. Coś takiego mogło być przełknięte przez MIT, ale nigdy przez Harvard. W czasie nauki w junior school matka zwróciła się o pomoc do terapeuty, który polecił skupić się na niechęci do pisania wypracowań i na wymądrzaniu się przy nauczycielach. Syn miał predyspozycje intelektualne do studiowania na Harvardzie, ale nie było pewne, czy starczy mu cierpliwości, by tak jak inni koledzy pisać na czas okresowe prace. Terapeuta zaproponował zrobienie próby. Jeżeli Richard przebrnie rok nauki w nowojorskiej szkole publicznej, w której jest wymagane pisanie prac okresowych z angielskiego, prawdopodobnie da sobie radę również na Harvardzie. Po zakończeniu nauki w junior high school Stallman zapisał się na letnie zajęcia prowadzone w Louis D. Bradeis High School na 84-tej ulicy i zaczął się uczyć trudnej sztuki pisania obowiązkowych prac klasowych, czego unikał w swojej dotychczasowej karierze szkolnej.

Jesienią powrócił w szeregi zwykłych uczniów szkoły. Ciężko mu przychodziło o cierpliwe przesiadywanie na lekcjach, które, w porównaniu z sobotnimi zajęciami w Columbia University, wydawały mu się pokutą, ale matka wspomina z dumą, że zdołał się podporządkować.

"Do pewnego stopnia czuł się upokorzony, ale dał radę" - wspomina. "Graniczyło z cudem, że tylko raz wezwano mnie do szkoły. Nauczyciel matematyki narzekał, że Richard wciąż

przerywa lekcje. Spytał am, w jaki sposób. ŹOskarża mnie, że źle przeprowadzam dowody. Ma rację, ale właściwych rozwiązań nikt, prócz niego, by nie zrozumiał t".

Na koniec pierwszego semestru w Brandeis wszystko się ułożyło. Ocena A 96 z angielskiego wymazał a pętno wcześniejszych niepowodzeń i 60 punktów otrzymanych 2 lata temu. Podtrzymał dobrą opinię świetnymi ocenami z historii amerykańskiej, matematyki i mikrobiologii. Ukoronowaniem osiągnięć było 100 punktów z fizyki. Choć nadal stronił od innych, 11 miesięcy nauki w Brandeis zakończył 4. lokatą wśród 789 rówieśników.

[Rysunek 0301]

Kopia świadectwa Stallmana z Louis D. Brandeis High School z listopada 1969. Należy zwrócić uwagę na doskonałą ocenę z angielskiego. "Do pewnego stopnia czuł się upokorzony, ale dał radę" - wspomina matka.

Poza szkołą uczył się z jeszcze większą niż dotychczas pilnością, wykonując swe obowiązki asystenta laboratoryjnego w Rockefeller University i uczęszczając na sobotnie zajęcia w Columbia University, gdzie po drodze musiał mijać protestujących przeciw wojnie w Wietnamie. Wszyscy biorący udział w Science Honors Program od dawna dyskutowali nad wyborem uczelni, aż wreszcie przyszedł moment, gdy i on musiał się zdecydować.

Breidbart wspomina: "Większość z nas wybierał a się, oczywiście, na Harvard lub MIT, choć było kilku decydujących się na inne uczelnie Ivy League. Z rozmów wynikało, że Richard jeszcze nie pisał słowa, gdzie zamierza studiować. Nie wiem, kto wreszcie zdobył się na odwagę, by spytać go o to".

Trzydzieści lat później Breidbart dokładnie pamięta ten moment. Po oświadczeniu Stallmana, że również wybiera się na

Harvard, w sali zapadła krępująca cisza. Wargi Richarda z wolna wgięły się w pełnym samozadowolenia uśmiechu. To było jego milczące stwierdzenie: "Tak... jeszcze się mnie nie pozbyliście".

Przypisy

1 Zobacz Michael Gross, "Richard Stallman: High School Misfit, Symbol of Free Software, MacArthur-certified geniusz" 1999. Jest to najszczerzy z wywiadów Stallmana. Polecam tę lekturę: <http://www.mgross.com/MoreThgsChng/interviews/stallman1.html>.

2 System edukacyjny w USA znacznie odbiega od naszego i trudno znaleźć zrozumiałe polskie odpowiedniki różnych stopni nauki. Dlatego zdecydowałem się podawać nazwy oryginalne, a orientację ułatwi zestawienie przygotowane na podstawie informacji ze strony <http://www.ed.gov/offices/OUS/international/USNEI/map.html>

Po lewej stronie mamy wiek dziecka, a po prawej lata nauki. Jak widać, są cztery systemy:

1. 8-4 - 8 lat szkoły podstawowej (elementary) i 4 lata średniej (high),
2. 4-4-4 - po 4 lata podstawowej, gimnazjum (middle) i średniej,
3. 6-3-3 - 6 lat podstawowej, 3 niższej średniej (junior high) i 3 wyższej średniej (senior high),
4. 6-6 - 6 lat podstawowej i 6 lat średniej (combined high).

Obecna wzmianka o middle school zdaje się świadczyć o edukacji według systemu 4-4-4, ale z dalszego tekstu wynika, że Stallman uczęszczał do junior high school i senior high school, co wskazywałoby na system 6-3-3. Ponieważ w trakcie nauki zmieniał szkołę, możliwe jest wymieszanie obu systemów, co zresztą nie jest zbyt istotne dla biegu opowiadania - przyp.

tl um.

3 Legendarny gracz w futbol amerykański. Patrz:

<http://www.cmgww.com/football/unitas/> - przyp. tl um

4 Patrz: pierwszy przypis w tym rozdziale - przyp. tl um.

5 Są to sobotnie wykłady organizowane przez Columbia University dla młodzieży szczególnie uzdolnionej w dziedzinie matematyki i nauk ścisłych. Więcej informacji można znaleźć pod adresem: <http://www.columbia.edu/cu/shp/>

- przyp. tl um.

6 Geek w slangu to osoba nierozgarnięta, niezręczna i nieprzystająca do innych lub tak zajęta jakimś tematem lub działaniem (również naukowym), że świata poza tym nie widzi. Stąd "Geek Syndrome" jako określenie autyzmu

- przyp. tl um.

7 W dalszej części rozdziału autor podaje adres internetowy tego artykułu

- przyp. tl um.

8 Asperger Syndrome - na stronie

<http://asperger.republika.pl/linki.htm> zespół Aspergera został zdefiniowany jako autyzm plus inteligencja - przyp. tl um

9 Patrz: Judy Steel, Toronto Star, BUSINESS, (9 października 2000): C03.

Jego wizja wolnego oprogramowania i powszechnej współpracy mocno kontrastuje z naturą skłoną do silnej izolacji w życiu prywatnym. Podobnie jak ekscentryczny kanadyjski pianista Glen Gould, jest błyskotliwy, wyrazisty i samotny. Uważa siebie za zarażonego w pewnym stopniu autyzmem, co utrudnia mu kontakty z innymi.

10 Nadmiernie nieuporządkowane zachowanie - przyp. tl um.

11 Patrz: Steve Silberman, The Geek Syndrome, Wired (grudzień 2001):

http://www.wired.com/wired/archive/9.12/aspergers_pr.html.

12 Żałuję, że pisząc tę książkę, nie mogłem przeprowadzić wywiadu

z Danielem Stallmanem. Gdy zabierał em się do jej pisanie, Richard Stallman poinformował mnie, że jego ojciec cierpi na chorobę Alzheimera. Gdy kończył em przygotowania w końcu roku 2001, dowiedziałem się, że Daniel Stallman zmarł w tym roku.

13 Patrz <http://www.madmag.com> - przyp. tłum.

14 To stwierdzenie na pewno wywołałoby sprzeciw ateisty Stallmana, ale na pewno było to coś, czego bardzo oczekiwał. W cytowanym już w przypisach wywiadzie udzielonym Michaelowi Grossowi mówił: "Gdy tylko dowiedziałem się o istnieniu komputerów, chciałem któryś zobaczyć i bawić się nim".

15 Liga bluszczowa - skupia najlepsze renomowane uczelnie amerykańskie, tak stare, że ich mury zdążyły yporosnąć bluszczem - stąd nazwa. Do Ligi bluszczowej należą Brown University, Columbia University, Cornell University, Dartmouth College, Harvard University University of Pennsylvania, Princeton University i Yale University - przyp. tłum.

16 Skala ocen w USA różni się od stosowanej w innych krajach i według informacji na stronach <http://www.educationguide-usa.com/isg/edusystem.htm> oraz <http://www.path2usa.com/KidsEdu.htm> wygląda następująco: - przyp. tłum.

4. IMPEACH GOD1

Choć stosunki między nimi był ynapięte, Richard odziedziczył po matce istotną i wartą odnotowania cechę: pełne pasji zainteresowanie polityką.

Jednakże cecha ta objawił a się w pełni dopiero po dziesięcioleciach. Kilka pierwszych lat życia spędził ,jak teraz przyznaje, w "politycznej próżni"². Jak większość Amerykanów w czasach Eisenhowera, rodzina Stallmanów starał a się w latach pięćdziesiątych powrócić do normalnego życia utraconego w wojennych czasach lat czterdziestych.

"Ojciec Richarda i ja należeliśmy do partii demokratycznej i to

nam wystarczało" - mówi Alice Stallman, wspominając lata spędzone w Queens. - "Nie angażowaliśmy się ani w politykę lokalną, ani państwową".

Zacząło się to zmieniać w późnych latach pięćdziesiątych, po rozwodzie Alice i Daniela Stallmanów. Powrót na Manhattan był czymś więcej niż zwykłą zmianą adresu, była to również zmiana dotychczasowego stylu życia i wytrącenie z błędnego spokoju.

"Myślę, że moje spotkanie z polityką zaczęło się od wizyty w bibliotece publicznej, gdzie znalazłem tylko jedną książkę na temat rozwodów" - wspomina Alice. - "Biblioteka była ściśle nadzorowana przez Kościół katolicki, przynajmniej w Elmhurst, gdzie mieszkaliśmy. Myślę, że wówczas po raz pierwszy uświadomiłam sobie istnienie sił, które nieśpiesznie i z ukrycia sterują naszym życiem".

Wracając na manhattański Upper West Side, w okolice znane z dzieciństwa, była zaskoczona zmianami, które zastała od czasu, gdy przed piętnastu laty opuściła Hunter College.

Powojenny głód mieszkaniowy zamienił okolicę w arenę walk politycznych. Z jednej strony byli prorozwojowi politycy z rady miejskiej dążący do przebudowy i rozbudowy kwartałów miasta i zasiedlenia ich tłumem napływających do miasta urzędników. Po drugiej stronie stała biedota irlandzkich i portorykańskich dzierżawców i podnajemców, która w okolicy znalazła nędzną przystań.

Na początku Alice Lippman nie wiedział, po której stronie ma stanąć. Jako świeżo zamieszkała w tej okolicy rozumiała, czym jest chęć posiadania mieszkania. Jako samotna matka z minimalnymi dochodami podzielała punkt widzenia biedaków zaniepokojonych planami budownictwa przeznaczonego dla bogatych. Oburzenie skłoniło ją do szukania sposobów walki z machiną polityczną dążącą do zamiany sąsiedztwa w kopię Upper East Side.

Wspomina swoją pierwszą wizytę w lokalnej siedzibie Partii Demokratycznej w roku 1958. Szukała instytucji, gdzie idąc do pracy, mogłaby pozostawiać syna, ale przeraziły ją warunki

panujące w jednym z utrzymywanych przez miasto ośrodków pomocy dla ubogiej ludności. "Wszystko, co pamiętam, to stęchły zapach skwaśniałego mleka, ciemne zakamarki i ogólny niedostatek. Byłam nauczycielką w prywatnej szkole opiekunek do dzieci. Kontrast był tak wielki, że po zajrzeniu do środka uciekłam. Bardzo mnie to wzburzyło".

Wizyta w lokalnej siedzibie partii, którą zapamiętała jako "pomieszczenie przesiąknięte zapachem tytoniowego dymu" przyniosła rozczarowanie. Jak wspomina, po tej wizycie uświadomiła sobie, że korupcja w partii może być rzeczywistą przyczyną wrogości miasta wobec ubogich. Nie zajął się do siedziby władz partyjnych, gdyż uznał, że lepiej przyłączyć się do jednego z klubów działających na rzecz zreformowania maszyny partyjnej demokratów i usunięcia z niej wpływów Tammany Hall³. Zapisła się do klubu, który szedł w ślady Woodrow Wilsona⁴/FDR⁵ Reform Democratic Club i wraz z innymi członkami zaczęła być na różnych zebraniach dotyczących planów zagospodarowania miasta oraz na zebraniach rady miejskiej, domagając się większej jawności działań.

"Naszym podstawowym celem była walka z Tammany Hall, Carminem DeSapio⁶ i jego pacholkiem" - wspomina.

"Reprezentował ten klub w radzie miejskiej i brał aktywny udział w przygotowaniu realnych planów zagospodarowania terenów miasta, które byłyby czymś więcej niż planem budowy kolejnych domów dla zamożnej ludności".

Ta działalność zaowocowała nasileniem działalności politycznej w latach sześćdziesiątych. W roku 1965 otwarcie wspierała kandydatów do władz politycznych z ramienia Partii Demokratycznej, wśród nich wybranego do Kongresu Williama Fittsa Ryana, który jako pierwszy z reprezentantów wypowiedział się przeciw wojnie w Wietnamie.

Wkrótce ona sama stała się jawną kontestatorką polityki USA w Indochinach. "Byłam przeciwniczką wojny w Wietnamie od czasu, gdy Kennedy posłał tam nasze wojska" - mówi. -

"Czytał am informacje prasowe i historie opisywane przez dziennikarzy we wczesnych fazach konfliktu. Uwierzył am ich zapowiedziom, że pakujemy się w bagno".

Takie opinie przenikały do domu. W roku 1967 matka Richarda ponownie wyszła za mąż. Nowy mąż, Maurice Lippman, szef Air National Guard⁷, zrezygnował ze stanowiska dla zademonstrowania postawy antywojennej. Jej pasierb, Andrew Lippman, studiował na MIT i korzystał ze studenckiego odroczenia służby, ale wisiała nad nim groźba powołania w przypadku eskalacji działań i sięgnięcia USA po kolejne zasoby ludzkie. Był wreszcie Richard, przed którym stanęła perspektywa wyjazdu do Wietnamu lub Kanady, gdyby wojna miała się przeciągnąć na lata 1970.

"Wietnam stał się głównym problemem naszego domu" - wspomina Alice Lippman. - "Rozmawialiśmy o tym bez ustanku: co pocniemy, jeżeli wojna będzie się przeciągać, co powinien zrobić Richard lub jego przyrodni brat w przypadku zaciągu. Byliśmy wszyscy przeciwnikami wojny i poboru. Uważaliśmy to za niemoralne".

Dla Stallmana wojna wietnamska była przyczyną wielu sprzecznych i złożonych uczuć: zakłopotania, przerażenia i głębokiej politycznej bezradności. Jako chłopak, który ledwo potrafił znaleźć swoje miejsce w autorytarnej rzeczywistości szkoły prywatnej, odczuwał dreszcz przerażenia na samą myśl o wdzianiu munduru.

"Strach mnie paraliżował i nie wiedział em, co począć, a jednocześnie nie mogłem się zdobyć na wzięcie udziału w demonstracjach antywojennych" - wspomina, jak na swoje, przypadające w marcu, 18 urodziny otrzymał przerażająco niski numer w loterii poborowej, gdy rząd w roku 1971 ostatecznie skasował odroczenia studenckie. "Nie mogłem sobie wyobrazić wyjazdu do Kanady lub Szwecji. W jaki sposób miał em się stąd zabrać i wyjechać? Nie wiedział em, jak można żyć, polegając jedynie na sobie. Nie należałem do osób, dla których takie rzeczy są proste i oczywiste".

Przypomina sobie, że był jednocześnie podbudowany i zawstydzony postawą członków rodziny. Poczynając od przyklejonej przez ojca nalepki na zderzaku, która porównywała masakrę w My Lai do podobnych okropności z czasów II wojny światowej. Był bardzo podekscytowany tym aktem odwagi ojca. "Podziwiałem to, co zrobił, i jednocześnie zdawałem sobie sprawę, że mnie nie byłoby na to stać. Obawiałem się, że porażający strachem potwór poboru dosięgnie mnie i zniszczy". Choć w opisach niechęci do wypowiadania własnych przekonań tkwi nuta zawstydzienia, od udziału w manifestacjach studenckich odstręczał Stallmana ton i kierunek ruchu antywojennego. Podobnie jak inni członkowie Science Honors Program postrzegał sobotnie manifestacje w Columbia University raczej jako zabawny spektakl⁸. Nie bez znaczenia był fakt, że irracjonalne siły kierujące ruchem antywojennym trudno było oddzielić od irracjonalnych podstaw kultury młodzieżowej. Wśród dziewcząt w wieku Stallmana uwielbienie dla Beatlesów przerodziło się w uwielbienie dla przywódców rewolty w rodzaju Abbiego Hoffmana i Jerry'ego Rubina. Chłopakowi usiłującemu zrozumieć swych rówieśników eskapistyczne slogany w rodzaju make love not war⁹ wydawały się mocno naciągane. Do Stallmana, krótko wystrzyżonego outsidera, nienawidzącego rocka i narkotyków, omijającego wszelkie odbywające się w kampusach demonstracje, nie przemawiał o nim tylko polityczne, lecz równie seksualne znaczenie hasła.

"Nigdy zbyt nie lubiłem kontrkultury" - przyznaje. - "Nie lubiłem muzyki. Nie lubiłem narkotyków. Do narkotyków miałem uciążliwość. Nie lubiłem postawy antyintelektualnej ani uprzedzeń do technologii. Ponad wszystko kochałem komputery. Nie cierpiałem również bezmyślnego antyamerykanizmu, z którym często się spotykałem. Wielu ludzi myślało o wszystkim tak uproszczony, że uważali, iż każda dezaprobata dla poczynań amerykańskich w Wietnamie musi pociągać za sobą wspieranie Wietnamczyków z północy. Nie

mogli sobie wyobrazić bardziej złożonego punktu widzenia". Takie komentarze łagodziły wrażenie nieśmiałości. Podkreślają również cechę, która stanie się kluczem do zrozumienia późniejszego politycznego dojrzewania Stallmana. Jego polityczna pewność siebie była wprost proporcjonalna do osobistej. Przed rokiem 1970 niewiele do niego docierało spoza świata matematyki i nauk ścisłych. Jednakże znajomość matematyki okazała się być solidnym fundamentem pozwalającym na ocenę ruchu antywojennego za pomocą rozważań logicznych. Czyniąc to, zauważył brak logiki działań. Choć sprzeciwiał się wojnie w Wietnamie, nie widział żadnej przyczyny, która dezawuowałaby wojnę jako środek obrony wolności lub przestrzegania sprawiedliwości. Nie zmniejszało to dystansu oddzielającego go od rówieśników, a nawet go powiększało, postanowił więc wyniki swych przemyśleń zachowywać wyłącznie dla siebie.

W roku 1970, rozpoczynając studia w Harvardzie, pozostawił za sobą wieczorne rozmowy przy kolacji o polityce i wojnie w Wietnamie. Patrząc wstecz, to przejście z matczynego domu do akademika w Cambridge nazywa "ucieczką". Rówieśnicy, którzy byli świadkami tej "ucieczki", nie zauważyli u niego żadnych oznak świadczących o wyzwoleniu.

"Na początku pobytu w Harvardzie wyglądał bardzo mizernie" - twierdzi Dan Chess, jeden z kolegów z Science Honors Program, który też znalazł się na tej samej uczelni. - "Kontakty z innymi istotami ludzkimi sprawiały mu wyraźne trudności, a w Harvardzie nie mógł ich uniknąć. Panował tam wyjątkowo towarzyska atmosfera".

Aby łatwiej znieść odmianę życia, Stallman zagłębił się w matematyce i naukach ścisłych. Jak większość kolegów z Science Honors Program z łatwością zaliczył egzamin kwalifikacyjny na Math 55, legendarny kurs startowy dla żadnych wiedzy początkujących matematyków. "My, koledzy z SHP, stworzyliśmy trwały związek, rodzaj matematycznej mafii" - śmieje się Chess, twierdząc, że "Harvard był drobnostką w

porównaniu z SHP".

Łatwe zaliczenie Math 55 dla Stallmana, Chessa i innych absolwentów SHP mogło być powodem do dumy. Wiedza matematyczna, rozkładana zwykle na 4 lata nauki, został a tu skondensowana w dwusemestrowym kursie, który znieść mogli tylko najlepsi zakochani w tym przedmiocie. "Był a to zdumiewająca grupa" - wspomina David Harbater, wówczas jeden z członków "mafii", a dzisiaj profesor matematyki w Pennsylvania University. "Można prawdopodobnie stwierdzić, że w historii tej uczelni nie było nigdy grupy początkujących studentów studiujących z takim zapalem i tak zaawansowanych. "Ludzie robili okrągł eoczy, gdy mówił em, że przed końcem drugiego semestru dyskutowaliśmy między innymi o geometrii rozmaitości różniczkowych Banacha. Zwykle na wykładach nie wspomina się o tym przed końcem drugiego roku studiów".

Kurs rozpoczęło 75 studentów, którzy rychł o poczęli się wykruszać, aż pozostał o ledwo 20, z czego, jak twierdzi Harber "moż e 10 wiedział o dokładnie, co robi. Z tych dziesięciu ośmiu zostało w przyszł ości profesorami matematyki, jeden zajął się nauczaniem fizyki, a ostatni to Richard Stallman".

Jak wspomina Seth Breidbart, jeden z kursantów Math 55, Stallman wyróżniał się nawet w tym gronie.

"W pewien swoisty sposób ściślej od innych przestrzegał procedur i dyscypliny działania" - mówi Breidbart. - "Istnieje w matematyce standardowa technika, zwykle błędnie przez wszystkich stosowana. Jest to nadużywanie notacji funkcji podczas jej definiowania. Najpierw funkcję definiujemy, a potem sprawdzamy, czy definicja jest właściwa. On zrobił tak tylko za pierwszym razem. Potem definiował relację i sprawdzał , że jest ona funkcją. Przebieg dowodu był ,oczywiście, taki sam, ale używał właściwej terminologii. Był wł aśnie taki".

Od czasu Math 55 zaczął a się utrwal ać renoma jego bł yskotliwości. Zgadza się z tym Breidbert, zaś Chess u którego wyczuwa się wspomnienie współzawodnictwa, uważa, że

Stallman został uznany za najlepszego matematyka dopiero na następnym roku. "Stał o się to w czasie wykładów z analizy funkcji rzeczywistych, które mieliśmy na drugim roku" - twierdzi Chess, obecnie profesor matematyki w Hunter College. - "Pamiętam złożony dowód dotyczący miar zespolonych, o którym Richard powiedział, że jest to w zasadzie analogia do rachunku wariacyjnego. Wtedy po raz pierwszy zobaczyłem kogoś, kto potrafi rozwiązać problem w sposób naprawdę oryginalny i błyskotliwy".

Chess nie ma wątpliwości: rozwiązanie, które Stallman napisał na tablicy, robiło wielkie wrażenie. Dla chłopaka, który przywykł, że zawsze uważano go za najlepszego matematyka w klasie, to, co zobaczył, wyglądało jak wyrok. Wiele lat później powoli zdobył markę świetnego profesjonalnego matematyka, ale pamięta, że tamten dowód Stallmana odczuł jako drwinę z siebie i z innych.

"W matematyce nie trzeba samemu być najlepszym" - twierdzi, - "by od pierwszego rzutu oka poznać prawdziwy matematyczny talent. Gdyby Richard został matematykiem, byłby matematykiem najwyższego lotu".

Naukowym sukcesom Stallmana towarzyszył brak sukcesów towarzyskich. Choć inni członkowie matematycznej mafii z Math 55 wspólnie omawiali rozwiązania i problemy, on wolał pracować samotnie. Tak samo był oż życiem prywatnym.

Składając prośbę o zakwaterowanie w Harvardzie, wyraźnie wypowiedział swoje preferencje. "Powiedziałem, że najlepiej odpowiada mi współlokator, którego nie będzie można ani zobaczyć, ani usłyszeć, ani dotknąć" - wspomina. Harvardzkie biuro kwater wykazało się tu rzadką biurokratyczną przenikliwością, przydzielając mu jedyną jednopokojową kwaterę przeznaczoną dla studentów pierwszego roku.

Breidbart, jedyny z mafii matematycznej współlokator z pierwszego roku, wspomina że Stallman powoli uczył się kontaktowania się z innymi studentami. Wspomina, jak inni koledzy z internatu, będąc pod wrażeniem logiczności jego

rozumowania, coraz chętniej go słuchali podczas intelektualnych debat w stołówce lub innych pomieszczeniach ogólnego użytku.

"Wiedliśmy mnóstwo nieformalnych dyskusji o możliwościach rozwiązywania światowych problemów i o ewentualnych skutkach tych rozwiązań" - mówi Breidbart. - "Powiedzmy, że ktoś wynalazłby eliksir nieśmiertelności. Co byście zrobili? Jaki miał dby to wpływ na politykę? Gdyby dać go wszystkim, świat uległby przeludnieniu i wszyscy by wyginęli. Gdyby dostęp do eliksiru ograniczyć do wybranej grupy, na przykład udostępnić go obecnie żyjącym, ale dawać go ich dzieciom, nastąpiłoby podzielenie ludzkości na klasę wyższą i niższą. Richard był najlepszy w przewidywaniu ukrytych skutków i okoliczności ewentualnych decyzji."

Stallman żywo pamięta ówczesne dyskusje. "Zawsze podobał a mi się idea nieśmiertelności i zdumiewało mnie, że większości widział a tylko jej złe strony. W jaki sposób inaczej moglibyśmy zobaczyć, jak świat będzie wyglądać za 200 lat?" - pyta.

Choć był matematykiem i dyskutantem najwyższej klasy, konsekwentnie unikał otwartego współ zawodnictwa, które mogłoby utwierdzić jego wybitną pozycję. Pod koniec pierwszego roku w Harvardzie jawnie uchylił się od zdawania egzaminu Putnama, prestiżowego sprawdzianu otwartego dla wszystkich studentów matematyki w USA i w Kanadzie. Oprócz możliwości porównania własnych umiejętności z rówieśnikami egzamin Putnama¹⁰ jest istotnym narzędziem rekrutacji najlepszych. Zwyczajowo zdobywca największej liczby punktów ma otwartą drogę na dalsze studia na dowolnie wybranej uczelni, łącznie z Harvardem.

Egzamin Putnama, podobnie jak Math 55, jest trudnym testem merytorycznym. Wydaje się, że pytania, na które trzeba odpowiedzieć w ciągu sześciu godzin podzielonych na dwie trzygodzinne sesje, są obmyślane tak, by skutecznie odsiać ziarno od plew. Breidbart, mający za sobą również Math 55, wspomina go jako najtrudniejszy egzamin, jaki kiedykolwiek

przyszło mu zdawać. "Abyście mieli pojęcie, jakie to trudne" - mówi - "powiem tylko, że na możliwych 120 punktów dostałem trzydzieści parę, co wystarczył odo zajęcia 102 miejsca w kraju".

Breidbart zdziwiony nieprzystąpieniem do egzaminu Stallmana, uznawanego za najlepszego matematyka na roku, wraz z innym kolegą przyparł go do muru, żądając wyjaśnienia. "Powiedział, że obawiał się, że nie pójdzie mu dobrze". Twierdzą, że gdy napisali z pamięci zadania egzaminacyjne, Stallman rozwiązał wszystkie. "Przypuszczam, że przez "nie pójdzie dobrze" Stallman rozumiał zrobienie jakiegoś błędu i zajęcie drugiego miejsca" - twierdzi Breidbart.

Stallman pamięta ten epizod nieco inaczej. Mówi:

"Przypominam sobie, że przynieśli mi pytania i być może odpowiedziałam na część z nich, ale na pewno nie na wszystkie". Ale zgadza się z opinią Breidbarta, że strach był główną przyczyną nieprzystąpienia do egzaminu. Mimo iż często wykazywał intelektualną niedoskonałość rówieśników i wykładowców, nie cierpiał stawiania do otwartej rywalizacji. "Z tego samego powodu nie lubiłem szachów" - twierdzi. - "Gdy tylko przystępowałem do gry, przede wszystkim się bałem, że zrobię głupi ruch na samym początku. I, oczywiście, stawało się to samospełniającą się przepowiednią".

Jest rzeczą dyskusyjną, czy ten strach powstrzymał Stallmana przed zrobieniem kariery matematycznej. Pod koniec pierwszego roku studiów zainteresował się zagadnieniami z innych dziedzin. Programowanie komputerowe, fascynacja z ostatnich klas szkoły średniej, zaczynało powoli stawać się jego główną pasją. Podczas gdy inni studenci matematyki szukali intelektualnego oddechu na wykładach z historii i sztuk pięknych, on znajdował go w laboratorium komputerowym. Pierwsze zetknięcie z prawdziwym programowaniem w IBM New York Scientific Center pobudził ochotę pogłębiania wiedzy. "Pod koniec pierwszego roku studiów nabrałem wystarczająco dużo odwagi, by odwiedzać laboratoria komputerowe i patrzeć, co

mają do zaoferowania. Pytał em, czy mają dodatkowe egzemplarze podręczników i dokumentacji, które mógłbym spokojnie przeczytać w domu".

Studiując je, poznawał specyfikacje komputerów, porównywał z już znanymi i przygotowywał programy próbne, które zanosił do laboratorium wraz ze zwracanymi podręcznikami. Choć niektóre laboratoria wzdragały się przed zaakceptowaniem dziwnego chłopaka zjawiającego się z prosto ulicy i z dopuszczeniem go do pracy z maszyną, część rozpoznała i uznała jego kompetencje i pozwalała uruchamiać napisane programy.

Pewnego dnia, gdzieś pod koniec pierwszego roku studiów, usłyszał o specjalnym laboratorium w pobliżu MIT. Znajdowało się ono na dziewiątym piętrze nienależącego do kampusu budynku na Tech Square - nowo zbudowanego specjalnie dla badań nad zaawansowanymi technologiami. Wedle krążących pogłosek, laboratorium to zajmowało się sztuczną inteligencją - wówczas najnowszą dziedziną badań komputerowych i programistycznych.

Zaintrygowany, postanowił złożyć tam wizytę.

Podróż nie była długa, wszystkiego dwie mile spacerem lub 10 minut pociągiem, ale jak się przekonał, MIT i Harvard znajdowały się na przeciwnych biegunach tej samej planety. Labirynt splecionych budynków kampusu MIT stanowił yin dla yanga Harvardzkiego przestronnego stylu kolonialnego. To samo można było powiedzieć o studentach, dziwnym skupisku odmieńców niegdyś niedopasowanych do wymagań szkół średnich, znanych bardziej z predylekcji do wygłupów niż poważnych zainteresowań politycznych.

Ta yin-yangowa zależność rozciągała się także na AI Lab - laboratorium zajmujące się zagadnieniami sztucznej inteligencji. W przeciwieństwie do tego, do czego Stallman był przyzwyczajony w Harvardzie, nie było tu żadnego stopniowania dostępu studentów do komputera, żadnych list z wyznaczonymi terminami dostępu i nawet śladu atmosfery w stylu "patrz i nie

dotykaj". Zamiast tego wszystkiego zobaczył jedynie rząd terminali i modeli ramion, zapewne artefakty z jakiegoś niedawnego doświadczenia prowadzonego przez AI Lab. Choć było widoczne, że każdy może usiąść przy jednym ze stojących terminali, Stallman postanowił trzymać się powziętego planu. Gdy spotkał pracownika laboratorium, spytał, czy mają zapasowe instrukcje i podręczniki, które mógłby pożyczyć jako dociekliwy student. "Coś tam mieli, ale wiele rzeczy nie miał o jeszcze żadnej dokumentacji" - wspomina. "Oni byli przede wszystkim hakerami". Znalazł tam coś lepszego od podręczników - pracę. Choć nie pamięta, czego dotyczył pierwszy projekt, pamięta, że w następnym tygodniu zjawiał się ponownie w AI Lab, zasiadł przy terminalu i zaczął pisać program.

Patrząc wstecz, nie widzi nic dziwnego w chęci zaakceptowania widzianego po raz pierwszy, nieznanego osobnika z zewnątrz. "Tak wówczas postępowano" - mówi. - "Tak również postępuje się teraz. Przyjmuję kogoś do pracy, jeżeli widzę, że jest dobry. Po co mam czekać? Tylko sztywniacy naciskają na przestrzeganie reguł administracyjnych w każdym punkcie. Jeżeli ktoś jest dobry, nie trzeba go zmuszać do przechodzenia przez całą ażmudną i nudną procedurę przyjmowania do pracy; należy go posadzić przy komputerze i niech się zabiera do roboty".

Aby posmakować "biurokratycznego sztywniactwa", wystarczyło odwiedzić któreś z laboratoriów komputerowych Harvardu. Dostęp do komputerów był ustalany zgodnie z pozycją akademicką delikwenta. Jako student pierwszego roku Stallman musiał zwykle wpisywać się na listę oczekujących lub czekać do północy, gdy profesorowie i studenci robią, co mieli zrobić, i pójść spać. Czekanie nie było trudne ale frustrujące. Oczekiwanie na dostęp do publicznego terminala w tym samym czasie, gdy połowa terminali umieszczonych w gabinetach profesorów stał abezczynnie, wydawał o musie rzeczą pozbawioną wszelkiej logiki. Choć od czasu do czasu odwiedzał

laboratoria komputerowe w Harvardzie, wolał egalitarną atmosferę AI Lab. "Był to powiew świeżego powietrza" - mówi. - " W AI Lab bardziej zwracano uwagę na pracę niż na rangi". Wkrótce przekonał się, że zasada "pierwszy ten, kto pierwszy przyszedł" jest zasługą kilku osób czujnie pilnujących jej przestrzegania. Wielu z nich przeszło do AI Lab z projektu MAC12, finansowanego przez Departament Obrony projektu badawczego, który doprowadził do powstania pierwszego systemu operacyjnego z podziałem czasu. Kilku z nich było już legendą w świecie komputerowym, np. Richard Greenblatt, ekspert od Lispu¹³ i autor MackHacka, programu do gry w szachy, który upokorzył Huberta Dreyfusa, krytyka sztucznej inteligencji. Był też Gerard Sussman, autor oryginalnego programu HACKER sterującego robotem układającym klocki¹⁴, oraz Bill Gosper, szczególnie uzdolniony matematyk już od 18 miesięcy tkwiący w hakerskich zagadnieniach wywodzących się z filozoficznych implikacji komputerowej gry LIFE¹⁵. Członkowie tej ścisłej grupy nazywali siebie hakerami. Po pewnym czasie objęli tym określeniem również Stallmana. Najpierw jednak wpoili mu zasady "hakerskiej etyki". Bycie hakerem to coś więcej niż pisanie programów, to pisanie możliwie najlepszych programów, to przesiadywanie bez przerwy przy monitorze przez 36 godzin tylko po to, aby napisany program był możliwie najlepszy, to również posiadanie dostępu do możliwie najlepszych maszyn i informacji. Hakerzy mówią otwarcie o zmienianiu świata za pomocą oprogramowania i, jak nauczyli Stallmana, pogardzają wszystkim, co staje na drodze do zrealizowania tego wzniosłego celu. Przeszkodami stającymi na ich drodze są między innymi byle jakie programy, biurokracja i samolubne zachowanie części programistów. Nauczono go także, w jaki sposób należy twórczo obchodzić te przeszkody. Nauczono go otwierania zamków, sztuki potrzebnej do dostawania się do zamkniętych gabinetów profesorów i "uwalniania" stojących tam bezużytecznie terminali. W

przeciwieństwie do swoich rozpieszczonych odpowiedników z Harvardu, kadra naukowa MIT wiedziała, że nie należy terminali traktować jak prywatnej własności. Jeżeli któryś z nich o tym zapomniał, hakerzy szybko korygowali ten błąd. Jeżeli "błąd" miał tendencję do powtarzania się, wysyłano dość ostrą reprymendę. "Widziałem wózek z ciężkim metalowym cylindrem użyty do otwarcia drzwi jednego upartego profesora" - twierdzi Stallman¹⁶.

Takie pozbawione subtelności metody osiągały cel. Choć profesorowie i administratorzy przewyższali liczbą hakerów w stosunku dwa do jednego, w AI Lab kierowano się hakerską etyką. W czasie przybycia Stallmana obie grupy żyły w symbiozie. W zamian za utrzymywanie w ruchu komputerów i usuwanie wszelkich usterek oprogramowania hakerzy uzyskali pozycję uprzywilejowanych pupilów pracujących nad najlepszymi projektami. Często prowadziło to do istotnych ulepszeń sprzętu i oprogramowania. Jak nastolatki cieszące się przebudową i rozbudową samochodów ze złomowiska, hakerzy klecili i "wiązali drutem" komputery, bawiąc się tym jak dzieci. Nigdzie to naprawianie drobiazgów nie było tak widoczne jak w systemie operacyjnym pracującym w AI Lab na głównym minikomputerze PDP-6. System ITS (skrót od Incompatible Time Sharing - niezgodne dzielenie czasu), miał w swoją architekturę wbudowane cechy wynikające z hakerskiej etyki. Hakerzy zbudowali go jako protest przeciw oficjalnemu systemowi CTSS (Compatible Time Sharing System - system zgodnego dzielenia czasu) stworzonemu w ramach projektu MAC i odpowiednio go nazwali. W owym czasie hakerom wydawało się, że system CTSS jest zbyt restryktywny i ogranicza programistom możliwości poprawiania jego architektury według potrzeb. Zgodnie z rozpowszechnianą przez hakerów legendą, decyzja o budowie ITS miała swoje polityczne podłoże. W przeciwieństwie do CTSS zaprojektowanego dla IBM 7094, ITS został zbudowany z myślą o PDP-6. Zezwolenie hakerom na zbudowanie własnego systemu oznaczało, że tylko

oni będą mogli swobodnie korzystać z PDP-6. I rzeczywiście, gambit działał. Chociaż maszyna była wspólnością kilku wydziałów, ludzie z A. I. mieli ją wkrótce dla siebie¹⁷.

System ITS miał funkcje, na które systemy komercyjne musiały czekać jeszcze przez lata, np. wielozadaniowość, debugging i możliwości edycji pełnoekranowej. Używając go jako głównego systemu PDP-6, AI Lab mogło korzystać z pełnej niezależności od projektu MAC już przed przybyciem tam Stallmana.

Jako rasowy adept hakerski Stallman wkrótce zakochał się w ITS. Zakazany dla większości nowicjuszy system zawierał wiele wbudowanych cech będących doskonałą szkołą dla terminatorów hakerskich w rodzaju Stallmana.

"ITS miał bardzo elegancki mechanizm wewnętrzny pozwalający na badanie jednego programu przez inny." - twierdzi Stallman - "Można było analizować wszystkie stany programu w bardzo specyficzny ale przejrzysty sposób".

Korzystając z tego, mógł śledzić, w jaki sposób pisane przez hakerów instrukcje są wykonywane. Drugą wspaniałą cechą była możliwość "zamrożenia" działania programu między poszczególnymi instrukcjami. W innych systemach operacyjnych takie polecenie prowadziło do ogólnego bałaganu i zawieszenia systemu. W ITS pozwalało to na śledzenie działania programu krok po kroku.

"Jeżeli wydawał on polecenie zatrzymania, system zawsze zatrzymywał się w trybie użytkownika. Można go było zatrzymać między wykonaniem dwóch kolejnych instrukcji użytkownika i wszystko było ze sobą zgodne i zdatne do prześledzenia" - wspomina. - "Po wydaniu polecenia kontynuacji zadanie było dalej wykonywane bez zakłóceń. Ponadto można było zmieniać status działania tam i z powrotem, i wszystko zgodnie pracowało. Nigdzie nie pojawiały się żadne stany ukryte".

Przed końcem roku 1970 działalność hakerska w AI Lab stała się stałym elementem tygodniowego planu zajęć Stallmana. Od poniedziałku do czwartku uczęszczał na zajęcia w Harvardzie.

W piątek po południu jechał do MIT na weekend. Zwykle czas przybycia dopasowywał do pory wieczornego posiłku i z pięcioma lub sześcioma innymi hakerami pakował się do samochodu, by przejechać przez most harwardzki do sąsiedniego Bostonu i zjeść coś w chińskiej restauracji. Przez dwie lub trzy godziny dyskutowali o wszystkim - od ITS do wewnętrznej logiki języka chińskiego i stosowanego w nim zapisu piktograficznego. Potem wracali do MIT i do świtu pisali hakerskie programy.

Dla samotnika rzadko potrafiącego się dopasować do swych rówieśników w Harvardzie było to istotne doświadczenie: spotkanie z grupą ludzi dzielących jego zainteresowania komputerami, science-fiction i chińską kuchnią. "Pamiętam wiele świtów, które nas zastawały w drodze powrotnej z Chinatown" - wspominał z nostalgią 15 lat później podczas przemówienia w Szwedzkim Królewskim Instytucie Technologicznym. - "Oglądanie wschodu słońca było rzeczą piękną, dającą spokój przez cały dzień. To wspaniała pora, by pójść spać. Wspaniale jest wracać do domu, gdy ptaki śpiewają. Daje to wspaniałe uczucie łagodnej, pełnej spokoju satysfakcji i zadowolenia z pracy wykonanej nocą¹⁸".

Im dłużej przebywał z hakerami, tym bardziej przesiąkał ich wizją świata. Od dawna przekonany do idei wolności osobistej, zaczął rozważać swe poczynania z punktu widzenia społecznej odpowiedzialności. Szybko zaczął oponować, gdy inni naruszali zasady społecznego współżycia. W rok od swego przybycia był jednym z włamujących się do zamkniętych pokoi, w których stały "zasekwestrowane" terminale będące przecież wspólną własnością całej społeczności. Zgodnie z hakerskim stylem chciał mieć swój udział we włamaniach do zamkniętych gabinetów. Jeden z najbardziej artystycznych sposobów otwierania zamkniętych drzwi, którego wymyślenie zwykle przypisuje się Greenblatowi, polegał na wygięciu twardego drutu na kształt łaski i przytwierdzeniu do dłuższego końca pętli z taśmy klejącej. Po wsunięciu drutu pod drzwiami można

go było obracać, trzymając za krótszy koniec, aż dłuższy koniec dotknął do gałki zamka. Wówczas, dzięki przytwierdzonej na końcu taśmie klejącej można był okilkoma szybkimi obrotami otworzyć zamek.

Gdy Stallman spróbował tego triku, uznał go za dobry ale wymagający kilku ulepszeń. Przytwierdzenie taśmy do końca pręta nie zawsze było łatwe, a obracanie prętem tak, by spowodować otwarcie zamka również wymagało sporej zręczności. Pamiętał, że na suficie korytarza znajdował się płyty, które można było łatwo odsunąć. Niektórzy hakerzy rzeczywiście korzystali z tej drogi, by dostać się do zamkniętych pomieszczeń, co dawało efekty lecz powodowało pokrycie delikwenta kurzem.

Stallman pomyślał o rozwiązaniu alternatywnym. Zamiast wsuwania pręta pod drzwiami należy usunąć płytę sufitową nad nimi, by móc spojrzeć na gałkę zamkniętego zamka od góry.

Wypróbował to. Zamiast drutu przygotował długą pętlę z taśmy magnetycznej w kształcie litery U. Na dole tego U przymocował pętlę z taśmy klejącej. Stojąc nad gałką zamka, tak długo manewrował pętlą, aż udało mu się umieścić ją pod gałką.

Wówczas podciągnął pętlę tak, by taśma klejąca przywarła do gałki zamka. Potem pociągnął lewy koniec taśmy magnetycznej, obracając gałkę zgodnie z kierunkiem ruchu wskazówek zegara. Drzwi się, oczywiście, otworzyły. Przyczynił się zatem do rozwinięcia sztuki otwierania zamkniętych drzwi.

"Czasami po przekręceniu gałki trzeba było popchnąć drzwi" - mówi, wspominając uporczywą trudność towarzyszącą korzystaniu z jego metody. - "Do otwarcia potrzebny był lekki nacisk".

Te poczynania wskazują na wzrastającą w Stallmanie chęć występowania w obronie przekonań politycznych. Panujący w AI Lab duch bezpośredniego działania pomógł mu przełamać wrodzoną strachliwość paraliżującą go w latach dzieciennych. Włamanie do gabinetu z zamkniętym terminalem

komputerowym nie był o tym samym co udział w marszu protestacyjnym, ale w przeciwieństwie do wielu marszów przynosiło efekty. Rozwiązywał o bieżący problem.

Przed dotarciem do ostatnich lat na Harvardzie zaczął w życiu uczelnianym stosować dziwaczne i nieakceptowane praktyki poznane w AI Lab.

"Czy opowiadał o wężu?" - spytała jego matka podczas przeprowadzanego z nią wywiadu. - "On i koledzy z internatu wystawili w wyborach węża. Oczywiście, zdobył sporo głosów". Stallman potwierdza opowiadanie o węzowej kandydaturze, dodając kilka uwag. Wąż był kandydatem w wyborach w jego internacie Currier House, a nie w wyborach do rady uczelni. Zdobył sporo głosów, dlatego że tak się nazywał jego właściciel. "Ludzie więc głosowali, myśląc, że chodzi o właściciela" - wspomina - "Na wyborczych plakatach podaliśmy, że wślizguje się do biur, a przed tygodniem wlaź do przewodu wentylacyjnego w ścianie i ukrył się tak, że nikt nie wiedział, gdzie się podziewa".

Wprowadzenie węża do rady internatu było tylko jednym z żartów. W następnych wyborach wprowadzili do rady internatu syna zarządcy, którego hasłem wyborczym było obowiązkowe przechodzenia na emeryturę w wieku 7 lat. Ale te żarty bladeły w porównaniu z kandydaturą kota Woodstocka do rady campusu. Woodstock zdobył wiele głosów, bijąc na głowę wielu ludzkich kontrkandydatów. "Nigdy nie ujawniono, ile głosów padło na Woodstocka. Głosy te unieważniono, ale sądząc z pokaźnej liczby nieważnych głosów w tamtych wyborach, można wnosić, że Woodstock był zwycięzcą". Kilka lat później Woodstock został w podejrzanych okolicznościach przejechany przez samochód. Nikt nie wiedział, czy kierowca pracował dla administracji MIT". Stallman twierdzi, że nie miał nic wspólnego z wystawieniem kandydatury Woodstocka, ale pomysł strasznie mu się podobał 19.

W AI Lab dział almuś polityczna Stallmana nabrała ostrzejszych tonów. W latach siedemdziesiątych hakerzy byli zmuszeni

odpierać ciągły nacisk władz i administracji uczelni, dążących do usunięcia przyjaznego dla hakerów systemu ITS. Jeden z pierwszych ataków przyszedł w połowie dekady, gdy coraz większa liczba kadry naukowej zaczęła się domagać bezpiecznego systemu operacyjnego z wiarygodną ochroną danych. Większość pozostałych laboratoriów komputerowych zainstalowała takie systemy w latach sześćdziesiątych, ale AI Lab pod naciskiem Stallmana i pozostałych hakerów utrzymywało obszar wolnego dostępu.

Dla Stallmana protest przeciwko zabezpieczeniom miał znaczenie zarówno etyczne, jak i praktyczne. Od strony etycznej wskazywał, że cała sztuka hakerska opiera się na intelektualnej otwartości i zaufaniu. Ze strony praktycznej twierdził, że wewnętrzna struktura ITS była zaprojektowana z myślą o otwartości i wszelkie odstępstwa od tego wymagają generalnej przebudowy.

"Hakerzy, którzy napisali Incompatible Timesharing System, uznali, że ochrona bezpieczeństwa plików jest potrzebna głównie samolubnym zarządcom systemów, którzy chcą w ten sposób zdobywać władzę nad innymi" - tłumaczył. - "Hakerzy nie chcą takiej władzy, więc nie wyposażyli systemu w narzędzia, które mogłyby jej służyć. W rezultacie, jeżeli tylko w systemie zdarzy się coś złego, można to łatwo naprawić²⁰".

Taka czujność hakerów pozwoliła na utrzymanie komputerów AI Lab poza systemami zabezpieczeń o wiele dłużej niż w innych laboratoriach MIT, ale w końcu zwyciężyli zwolennicy ochrony danych. W roku 1977 MIT Laboratory for Computer Sciences²¹ zainstalowało pierwszy system z ochroną danych za pomocą haseł. Stallman ponownie uznał za swoje zadanie napiętnowanie tego, co uważał za odstępstwo od zasad etycznych. Zdobył dostęp do programu nadzorującego system haseł i zainstalował polecenie wysyłające komunikat do każdego, kto usiłował założyć unikalne hasło. Jeżeli użytkownik wpisał hasło np. "rozgwiazda", otrzymywał komunikat takiej mniej więcej treści:

Widzę, że wpisałeś hasło "rozwiazda". Proponuję, abyś je zmienił na proste naciśnięcie "Enter". To o wiele prostsze i popierające zasadę nieużywania żadnych haseł²².

Użytkownicy, którzy posłuchali rady, czyli nacisnęli Enter, wprowadzali do systemu pusty ciąg znaków, co było równoważne brakowi hasła i udostępnieniu swoich zasobów całemu światu. Niezależnie od tego, jak straszne skutki mogło to przynieść samym użytkownikom, umocniło wśród nich hakerską ideę, że komputery należące do Instytutu, a nawet zapisane na nich pliki są własnością powszechną a nie prywatną. Stallman, wypytywany o to w roku 1984 w wywiadzie związanym z przygotowywaniem książki Hakerzy, z dumą oświadczył, że jedna piąta użytkowników LCS uznała podany argument i wprowadził apuste hasła²³.

Jednakże jego walka o puste hasła okazała się daremna. Do roku 1980 nawet komputery AI Lab korzystały już z systemów używających hasłowego systemu zabezpieczeń. Był to kamieniem milowym na drodze jego osobistego i politycznego dojrzewania. Dla obiektywnego obserwatora jego późniejszej kariery jest to wygodny punkt graniczny oddzielający czas nieśmiała egochłopca obawiającego się wyrażać swoje opinie od czasu dorosłego aktywisty, który wkrótce poświęci całą energię na przekonywanie i nakłanianie innych.

Sprzeciwiając się wprowadzeniu systemu zabezpieczeń komputerowych, Stallman wykorzystał wiele sił, które kierowały jego postępowaniem w pierwszym okresie życia: głód wiedzy, niechęć do autorytetów i frustrację wywołaną przez wszelkie ukryte procedury czyniące z wielu ludzi bezideowców. Skorzystał także z koncepcji etycznych, które kierowały jego dorosłym życiem: społecznej odpowiedzialności, zaufania i hakerskiej zasady bezpośredniego działania. W języku komputerowym można powiedzieć, że puste hasło było wersją 1.0 Richarda Stallmana-działacza; wersją jeszcze niekompletną i pełną braków, ale dojrzłą wzarysie.

Patrząc wstecz, Stallman waha się, czy temu zdarzeniu z

okresu hakerskiego można przypisać tak duże znaczenie. "W tym czasie" - mówi - "wielu podzielało moje poglądy. Świadczy o tym znaczna liczba osób, które zaakceptował ypropozycję wprowadzenia pustego hasła. Po prostu miał em skłonności do stania się dział aczem, obrońcą tej sprawy".

Przyznaje jednak, że to AI Lab uświadomiło mu, w jaki sposób można wyrażać swoje przekonania. Jako nastolatek obserwował wiele zdarzeń politycznych, ale nie przypuszczał ,by pojedynczy osobnik mógł zrobić coś, co miałoby jakiegokolwiek znaczenie.

Jako dorosły y mł dzeniec mówił o rzeczach, których był p ewien, takich jak projektowanie oprogramowania, odpowiedzialność społeczna i wolność osobista. "Wł ączył em się do społeczności, która respektowała wolność każdego czł onka i w krótkim czasie uświadomiłem sobie, jaka to ważna rzecz. Dopiero po pewnym czasie zrozumiałem, że jest to sprawa natury moralnej" - mówi.

Hakerstwo w AI Lab nie było jedyną formą działalności pomagającą mu w zdobyciu prestiżu. Na drugim roku przył ączył się do zespołu tanecznego specjalizującego się w tańcach ludowych. To, co miał o być jedynie okazją do spotykania kobiet i rozrywką towarzyską, rychło stało się jego drugą obok hackerstwa pasją. Tańcząc przed publicznością w stroju bałkańskiego wieśniaka, tracił wcześniejszą nieporadność dziesięcioletniego chłopca, dla którego nieudane próby gry w piłkę był yprzyczyną frustracji. Zdobył pewność siebie, sprawność fizyczną i chęć życia. Przez krótką chwilę czuł nawet związek emocjonalny z innymi. Wkrótce przekonał się, że występowanie przed publicznością może być świetną zabawą i wkrótce strona artystyczna przedstawienia zaczęła go interesować na równi ze stroną towarzyską.

Taniec i hackerstwo nieco poprawił yjego pozycję towarzyską, pomógł ypokonać uczucie odmienności, które ciemną chmurą przysłaniało jego przedharvardzkie życie. Zamiast lamentować nad swoją dziwną naturą, zaczął ją celebrować. W roku 1977, biorąc udział w zjeździe miłośników science-fiction, natknął się na kobietę sprzedającą guziki robione na

zamówienie. Podekscytowany zamówił guzik z napisem Impeach God²⁴.

Dla niego było to przesłanie o wielu poziomach znaczeniowych. Jako ateista od najwcześniejszego dzieciństwa uznał to przede wszystkim za wezwanie do stworzenia drugiego frontu w toczącej się debacie religijnej. "Dotychczas wszyscy argumentowali za istnieniem lub nieistnieniem Boga" - twierdzi, - "a hasło Impeach God wzywa do spojrzenia na Boga od zupełnie innej strony. Jeżeli Bóg był na tyle potężny, by stworzyć świat, a potem nie zrobił nic, by poprawić swe dzieło, dlaczego mamy go czcić, zamiast postawić w stan oskarżenia? Czy nie lepiej byłoby wytoczyć mu proces?"

Jednocześnie Impeach God było hasłem satyrycznym uderzającym w Amerykę i jej system polityczny. Skandal polityczny afery Watergate z lat siedemdziesiątych głęboko dotknął Stallmana. Jako dziecko wzrastał w atmosferze niedowierzania autorytetom. W życiu dorosłym ta niewiara została awzmocniona kulturą hakerskiej społeczności AI Lab. Dla hakerów afera Watergate była jedynie jedną z szekspirowskich realizacji codziennych walk, które przychodzi staczać ludziom pozbawionym przywilejów z potęgami świata. Była to wielka przenośnia tego, co się dzieje, gdy ludzie zaprzędają wolność i otwartość dla wygody i bezpieczeństwa. Rosnąca pewność siebie skłoniła go do dumnego noszenia owego guzika. Ciekawskim pytającym o znaczenie hasła odpowiadał dobrze spreparowanym oświadczeniem: "Moje imię Jehowa. Mam plan zbawienia wszechświata, ale ze względu na niebiańskie zasady bezpieczeństwa nie mogę więcej powiedzieć. Możecie tylko wierzyć we mnie, ponieważ ja widzę obraz, a wam nie jest to dane. Wierzcie w moją dobroć, ponieważ mówię wam o niej. Jeżeli mi nie uwierzycie, wpiszę was na listę moich wrogów i wrzucę tam, gdzie piekielni poborcy podatkowi rozliczą was ze zobowiązań wobec wieczności".

Ci, którzy odczytywali to przemówienie jedynie jako parodię informacji o aferze Watergate, chwytały jedynie połowę

znaczenia. Według Stallmana dodatkowa informacja zawarta w tych słowach był a przeznaczona jedynie dla innych członków hakerskiej społeczności. W sto lat po ostrzeżeniu lorda Actona²⁵, że władza absolutna korumpuje absolutnie, Amerykanie zdają się zapominać o pierwszej części wygłoszonego przez niego truizmu: władza to korupcja. Zamiast zajmować się poszczególnymi przypadkami korupcji Stallman swą nienawiść skierował na system, który potęgę władzy stawiał na pierwszym miejscu.

"Pytał em, dlaczego mamy się zatrzymać na pomniejszym wyrobniku" - mówi Stallman, wspominając guzik i zapisane na nim przesł anie. - "Skoro uporaliśmy się z Nixonem, dlaczego nie zająć się Najwyższym? Sądę, że każdy, kto mając władzę, używa jej do poniżenia innych, powinien zostać jej pozbawiony".

Przypisy

1 Można to przetł umaczyć jako "postaw Boga w stan oskarżenia", ale wydźwięk polskiego opisowego tł umaczenia jest nieco inny niż oryginał u. Dlatego zdecydowałem się na pozostawienie oryginalnego tytuł u opatrzonego przypisem, tym bardziej, że po aferze "rozporkowej" prezydenta Clintona ludzie są osłuchani ze słowami "impeach" i "impeachment", bowiem prasa ciągle je przytaczał a i odmieniał a na wszystkie możliwe sposoby - przyp. tł um.

2 Patrz Michael Gross, Richard Stallman: High School Misfit, Symbol of Free Software, MacArthur-certified Genius (1999).

3 The Tammany Society of New York City powstało w roku 1786 jako towarzystwo wzajemnego wsparcia mieszkańców Nowego Jorku. Najpierw zajmowało się głównie opieką socjalną, ale od roku 1798 zaczęło się upolityczniać. Przez 80 lat od roku 1854 do 1954 faktycznie rządziło miastem przez agendy podporządkowanej sobie lokalnie partii demokratycznej. Dla nowojorczyków stało się synonimem przekupstwa i ciemnych

interesów władz miejskich. Dominacja Tammany Hall został a
złamana w roku 1954, ale znaczne wpływy istniały jeszcze w
latach sześćdziesiątych - przyp. tłum.

4 Woodrow Thomas Wilson (1856 - 1924), 28. prezydent
USA, sprawował tę funkcję przez dwie kadencje w latach 1913 -
1921. Realizował program tzw. Nowej Wolności (New Freedom),
obejmujący reformy gospodarcze, prawne i administracyjne.
Zreformował system bankowy (1913) i wprowadził
ustawodawstwo antymonopolowe (1914), wolność strajku i
zrzeszania się, walczył z korupcją. Przyp. tłum. na podstawie
<http://wiem.onet.pl/>

5 Franklin Delano Roosevelt (1882 - 1945). Działacz Partii
Demokratycznej, w latach 1933-1945 czterokrotnie wybierany
na prezydenta USA. Twórca programu reform społecznych
(m.in. wprowadzenia ubezpieczeń dla bezrobotnych, rent, płac
minimalnych) i ekonomicznych, opartych na zasadzie
interwencjonizmu państwowego (New Deal). Przyp. tłum. na
podstawie <http://wiem.onet.pl/>

6 Carmine DeSapio miał wątpliwy zaszczyt bycia pierwszym
włoskiego pochodzenia szefem Tammany Hall, nowojorskiej
machiny politycznej. Więcej informacji na temat DeSapio i
politycznych rozgrywek w Nowym Jorku po II wojnie światowej
można znaleźć w "Spinning the Niger: Carmine DeSapio end
the End of the Tammany Era", "New York Affairs" (1975):3:1.

7 US Air National Guard - Lotnictwo Gwardii Narodowej
Stanów Zjednoczonych - przyp. tłum.

8 Chess, inny uczestnik Columbia Science Honors Program,
opisuje protesty jako "krzyki w tle". "Wszystkich nas
interesowała polityka, ale najważniejszy był SHP. Nigdy nie
opuszczaliśmy zajęć dla demonstracji."

9 "Rób miłość nie wojnę" - podaję wersję angielską i to
niezgrabne tłumaczenie, gdyż takie właśnie transparenty
widywałem również w Polsce - przyp. tłum.

10 Egzamin Putnama - William Lowell Putnam Mathematical
Competition

- jest co roku organizowany przez MAA (Mathematical Association of America) dla studentów wydziałów matematycznych uczelni amerykańskich i kanadyjskich. Upamiętnia Williama Lowella Putnama, który od roku 1927 propagował ideę zorganizowanego naukowego współzawodnictwa młodzieży. Egzamin Putnama odbywają się od roku 1938. Więcej informacji można znaleźć np. pod adresem <http://math.scu.edu/putnam/index.html> - przyp. tł um.

11 Yin i yang (jin-jang, in-jang) to wg tradycji chińskiej podstawowe pierwiastki-sił ytworzące rzeczywistość, decydujące o jej trwaniu i nieustannych przemianach. Zawsze w parze, choć w różnych proporcjach - są swoim dokładnym przeciwieństwem, ale równocześnie jedno zawiera w sobie cząstkę drugiego, co decyduje o ich nierozdzielności - przyp. tł um.

12 Projekt MAC (Multi Access Computer - komputer wielodostępowy) realizowany przez wiele lat w MIT, począwszy od roku 1962 (plany wstępne opracowano o rok wcześniej), ściągnął do Cambridge około 400 najlepszych specjalistów. Celem było stworzenie komputera, z którego mogłoby korzystać jednocześnie wiele osób, co pozwalałoby na uniknięcie opóźnień wynikających z oczekiwania przy stosowaniu procesu wsadowego, w którym każdy po kolei uruchamiał swój program, otrzymując wyniki w tej samej kolejności. Rezultatem prac prowadzonych w ramach tego projektu jest między innymi podział czasu procesora. Te rzeczy, tak dziś powszechne i oczywiste, kosztowały masę czasu, wysiłku i pieniędzy. Budżet projektu MAC w roku początkowym wynosił 2 mln dolarów, w najbardziej kosztownym roku 1969 - 4,3 mln dolarów, po czym spadł do około 3 mln dolarów rocznie. Opracowałem na podstawie kilku rozproszonych źródeł internetowych, więc nie podaję jednego adresu WWW. Zainteresowani mogą znaleźć więcej informacji za pomocą wyszukiwarki - przyp. tł um.

13 Lisp - język programowania wysokiego poziomu

wykorzystywany do badań nad sztuczną inteligencją. Choć należy do najstarszych języków programowania, nadal jest używany - przyp. tł um.

14 To z pozoru proste zagadnienie dotyczy fundamentalnej różnicy między ludźmi i komputerami: ludzie przez gromadzenie doświadczeń powiększają swą umiejętność rozumienia świata, czego komputery nie potrafią. Dlatego proste układanie klocków, z czym dziecko radzi sobie dzięki intuicji, jest dla komputerów bardzo trudnym problemem - przyp. tł um.

15 Patrz Steven Levy, Hackers (Penguin USA - 1984), str. 144.

Levy poświęca około 5 stron na opis zafascynowania Gospera grą LIFE, opartym na matematycznych podstawach programie stworzonym przez brytyjskiego matematyka Johna Conwaya. Szczerze polecam tę książkę jako lekturę uzupełniającą lub wstępną.

16 Przeczy temu Gerald Sussman, członek kadry naukowej MIT i był yhaker z AI Lab. Wedł ug niego, hakerzy nigdy nie wł anywali się do pomieszczeń z terminalami.

17 Przepraszam za skrótowe omówienie powstania ITS, systemu, który niektórzy hakerzy do dziś uważają za wcielenie hakerskiego etosu. Więcej informacji na temat politycznego znaczenia tego programu można znaleźć w: Simson Garfinkel, "Architects of the Information Society: Thirty Five Years of the Laboratory for Computer Science at MIT" (MIT Press, 1999).

18 Patrz: Richard Stallman "RMS lecture at KTH (Szwecja)", (30 października 1986):

<http://www.gnu.org/philosophy/stallman-kth.html>.

19 W czasie, gdy ta książka była w ostatniej fazie przygotowania do druku, otrzymał em ę Stallmana e-mail, w którym wyjaśniał , że to campus Harvardu wzbudził jego zainteresowanie polityką. "W czasie mojego pierwszego roku w Harvardzie na zajęciach z historii Chin przeczytał em historię pierwszej rewolty przeciw cesarzowi z dynastii Tsin" - pisze.

"Nie było to wiarygodne źródło historyczne, ale historia bardzo mnie poruszył a".

20 Patrz: Richard Stallman (1986).

21 Laboratorium Nauk Komputerowych MIT - przyp. tł um

22 Patrz: Steven Levy, Hackers Penguin USA 1984, str. 417.

Nieco zmodyfikowałem ten cytat, a Levy również podał jedynie fragment, aby jaśniej pokazać, w jaki sposób Stallman ujawnił pozorną bezpieczeństwo ochrony za pomocą haseł. Levy zamiast konkretnego hasła podaje symbole.

23 Patrz Steven Levy, Hackers Penguin USA 1984, str. 417.

24 Patrz pierwszy przypis w tym rozdziale - przyp. tł um

25 Lord John Emerich Edward Dalberg Acton (1834 - 1902), brytyjski historyk, myśliciel polityczny i eseista. Jeden z założycieli The English Historical Review, od 1895 profesor historii nowożytnej uniwersytetu w Cambridge. Reprezentant liberalnego kierunku w katolicyzmie angielskim, oponującego przeciwko doktrynie o nieomyślności papieża w sprawach wiary i moralności. Krytykował rasizm, nacjonalizm i socjalizm. Zagrożeń dla wolności człowieka dopatrywał się w wypaczeniach systemu demokratycznego i centralizacji władzy państwowej. Według encyklopedii internetowej <http://wiem.onet.pl/> - przyp. tł um.

5. NIEWIELKIE BAJORKO WOLNOŚCI

Spytajcie kogokolwiek, kto spędził choć minutę ze Stallmanem, co najbardziej utkwiło mu w pamięci, a otrzymacie zawsze tę samą odpowiedź. Nie będą to ani długie włosy, ani dziwny sposób zachowania. Jego najbardziej zauważalną cechą jest wyraz oczu. Jedno spojrzenie w zielone oczy Stallmana i jesteście pewni, że macie przed sobą człowieka, który wierzy w to, co robi.

Intensywność jego wzroku nie podlega dyskusji. Jego oczy na Ciebie nie patrzą, one Cię przenikają na wskroś. Nawet jeżeli

przez grzeczność na chwilę odwracasz wzrok, jego oczy nie przestają Cię w skupieniu lustrować, wwiercając się w czaszkę jak dwie wiązki fotonów.

Być może dlatego większość pisarzy, opisując Stallmana, robi to pod kątem religijnym. W roku 1998 w Salon.com opublikowano artykuł zatytułowany "The Saint of Free Software"¹, w którym Andrew Leonard napisał, że "zielone oczy Stallmana promieniują mocą starotestamentowych proroków"². W roku 1999 magazyn Wired napisał, że broda Stallmana przypomina rasputinowską³, zaś London Guardian opisuje uśmiech Stallmana jako należący do "ucznia spoglądającego na Jezusa"⁴.

Te analogie robią wrażenie, lecz ostatecznie chybiają celu, dlatego że nie odkrywają słabych punktów Stallmana. Jeżeli dłużej popatrzyście mu w oczy, zauważycie subtelna zmianę. To, co na pierwszy rzut oka wydaje usiłowaniem onieśmielenia i zahipnotyzowania, okazuje się być próbą nawiązania i podtrzymania kontaktu przez osobę sfrustrowaną. Jeżeli osobowość Stallmana, zgodnie z jego własnymi podejrzeniami, jest wynikiem autyzmu lub syndromu Aspergera, jego oczy zdają się potwierdzać tę diagnozę. Nawet wówczas gdy błyszcza najintensywniej, zdają się patrzeć gdzieś w dal, jak oczy zranionego zwierzęcia, które ma wyzionać ducha.

Moje pierwsze zetknięcie z legendarnym spojrzeniem Stallmana nastąpiło w marcu roku 1999 na LinuxWorld Convention and Expo⁵ w San Jose w Kalifornii. Konferencja, dla społeczności związanej z oprogramowaniem linuksowym będąca "wyjściem w świat", Stallmanowi ułatwiała aponowne zaprezentowanie się mediom elektronicznym. Zdecydowany upomnieć się o należną mu część uwagi opinii publicznej, skorzystał z okazji, by poinformować zwiedzających i dziennikarzy o historii projektu GNU i jego politycznych celach.

Jako reporter posłany do obsługi zdarzenia poszedłem na konferencję Stallmana poświęconą pojawieniu się GNOME 1.0 - graficznego interfejsu użytkownika należącego do wolnego

oprogramowania. Zadając pierwsze pytanie, nacisnął em całą serię "czerwonych guzików", co spowodowało wygłoszenie długiego i wyczerpującego pouczenia. Zapytałem, czy dojrzała wersja GNOME będzie miała awpływ na rynkową popularność systemu operacyjnego Linux.

"Bardzo proszę nie nazywać systemu operacyjnym Linuksem" - odparł Stallman. - "Linuksowe jądro jest tylko niewielką częścią systemu operacyjnego. Znaczna część programów stanowiących razem system operacyjny, który nazywa pan Linuksem, nie został anapisana przez Linusa Torvaldsa. Programy te stworzyli ochotnicy pracujący w ramach projektu GNU i poświęcający na to swój prywatny czas. Dzięki temu mamy wolny system operacyjny w obecnej postaci. Nieuznawanie ich wysiłków jest niegrzeczne i fałszuje fakty historyczne. Dlatego proszę, aby pan, mówiąc o tym systemie operacyjnym, używał jego włściwej nazwy GNU/Linux".

Notując usłyszane słowa, wsłuchiwałem się w niesamowitą ciszę, która zapadła w zatłoczonej sali. Gdy wreszcie podniosłem wzrok, napotkałem oczy Stallmana wpatrujące się we mnie bez jednego mrugnięcia. Inny reporter nieco bojaźliwie zadał następne pytanie, pilnując się, by używać włściwej nazwy GNU/Linux. Odpowiedział na nie Miguel de Icaza kierujący projektem GNOME. Jednakże dopiero gdy de Icaza był w połowie odpowiedzi, Stallman wreszcie odwrócił wzrok ode mnie. Dopiero wtedy poczułem miły dreszcz wolności przebiegający po krzyżu. Gdy Stallman zaczął pouczać innego reportera na temat błędów jego dykcji, poczułem ulgę.

Przynajmniej już na mnie nie patrzy - pomyślałem.

Takie chwilowe zatrzymywanie na kimś wzroku doskonale służyło celom Stallmana. Przed końcem konferencji LinuxWorld większość dziennikarzy doskonale wiedział a, że w jego obecności lepiej nie używać nazwy Linux, a w wired.com pojawił się artykuł ,w którym porównano Stallmana do przedstalinowskiego rewolucjonisty wymazanego z kart historii przez hakerów i przedsiębiorców, którym z powodów

politycznych projekt GNU stał ością w gardle⁶. Autorzy innych artykułów poszli tym śladem i gdy kilku dziennikarzy użyło w druku nazwy GNU/Linux, inni przypomnieli, że Stallman doprowadził do powstania wolnego systemu operacyjnego już 15 lat wcześniej.

Przez następne 17 miesięcy nie spotkałem Stallmana. W tym czasie ponownie odwiedził Dolinę Krzemową podczas Linux-World Show w sierpniu 1999 r. Choć nie poproszono go o wygłoszenie przemówienia, urządził wszystko w sposób maksymalnie dla siebie korzystny. Przyjmując Linus Torvalds Award for Community Service⁷, zażartował: "Danie nagrody Linusa Torvaldsa Fundacji Wolnego Oprogramowania to coś jak nagroda Hana Solo dla Przymierza Rebeliantów⁸".

Tym razem jednak komentarz przeminął w mediach bez echa. W połowie tygodnia Red Hat, Inc., najważniejszy dostawca GNU/Linux, wszedł na giełdę z ofertą publiczną. Było to potwierdzenie czegoś, czego dziennikarze, łącznie ze mną, spodziewali się od dawna - "Linux" stał się na Wall Street słowem-symbolem, podobnie jak wcześniej e-commerce i dot-com. W miarę zbliżania się przesilenia roku 2000 na giełdzie wszystkie dyskusje na temat wolnego oprogramowania i jego politycznego znaczenia odbijały się od nieprzebitego muru, jak hiperbola od pionowej asymptoty, i opadały bezsilnie u jego podnóża.

Być może dlatego trzecią z kolei konferencję LinuxWorld w sierpniu 2000 r. Stallman demonstracyjnie zbojkotował. Wkrótce potem po raz drugi zetknąłem się z nim i z jego "firmowym" spojrzeniem. Słyszając, że zamierza przyjechać do Krzemowej Doliny, umówiłem się z nim na lunch i wywiad w kalifornijskim Palo Alto. Wybór miejsca zdawał się być ironią losu, nie tylko z powodu ostatniej nieobecności Stallmana na LinuxWorld, lecz także z powodu tła naszego spotkania. Poza Redmond w stanie Washington niewiele jest miast, które byłyby w równym stopniu potwierdzeniem ekonomicznego znaczenia własności oprogramowania. Ciekawie było patrzeć, jak

Stallman - człowiek, który spędził znaczną część życia na zwalczaniu mentalności opartej na chęci posiadania, zachowa się w mieście, w którym nawet cena bungalowu wielkości połowy garażu sięga 500 tysięcy dolarów. Jechałem na spotkanie z Oakland.

Kierując się wskazówkami otrzymanymi od Stallmana, dotarłem do siedziby Art.net, niedochodowej "wspólnoty wirtualnych artystów". Kwatera Art.net usytuowana na północnym krańcu miasta, na posesji odgradzonej od świata żywopłotem, działała odświeżająco. Nagle wizja Stallmana czającego się gdzieś w zakamarkach Krzemowej Doliny wydała się mniej dziwaczna.

Znalazłem go w zaciemnionym pokoju, "zastukanego" w klawiaturę jego szarego laptopa. Gdy wszedłem, podniósł wzrok i oświecił mnie pełną mocą swego 200-watowego spojrzenia. Gdy mruknął dość przyjazne "Hello", odwzajemniłem powitanie. Zanim jednak cokolwiek powiedzieliśmy, jego wzrok powrócił na ekran laptopa.

"Właśnie kończę artykuł poświęcony duchowi hakerstwa" - powiedział, nadal przebiegając palcami po klawiaturze. - "Zobacz".

Spojrzałem. Światło w pokoju było przyćmione, a tekst był wyświetlony białozielonkawym kolorem na czarnym tle, czyli za pomocą kolorów odwrotnych do zwykle stosowanych w edytorach tekstu, zatem nieco potrwało, zanim mój wzrok zaadaptował się do warunków i mogłem coś odczytać.

Zobaczyłem opis ostatniej wizyty Stallmana w restauracji koreańskiej. Przed posiłkiem Stallman dokonał interesującego odkrycia: osoba nakrywająca do stołu pozostawiła 6 pałeczek zamiast potrzebnych dwóch. Większość restauracyjnych gości po prostu zignorowałaby istnienie dodatkowych czterech pałeczek, ale dla niego było to wyzwanie: w jaki sposób użyć jednocześnie wszystkich sześciu. Tak jak w przypadku wielu hakerskich trików, również tutaj rozwiązanie było jednocześnie proste i wymagające inteligencji, więc użył ich tak, jak to

zostało pokazane na załączonej ilustracji.

W trakcie czytania tej historii czułem, że Stallman intensywnie mi się przygląda. Zerknąłem na niego i zauważyłem na twarzy dziecięcy półuśmiech zadowolenia z siebie. Ale gdy pochwaliłem jego esej, mój komentarz wywołał ledwo zauważalne uniesienie brwi.

"Będę gotowy za moment" - powiedział .

Wrócił do stukania na laptopie. Był to szary kanciasty laptop, w niczym nie przypominający lśniących, nowoczesnych cacek, które często widywałem w rękach programistów na konferencji LinuxWorld. Nad normalną klawiaturą była przymocowana druga pozwalająca na lżejsze naciskanie klawiszy. Gdy w latach osiemdziesiątych spędzał około 70-80 godzin tygodniowo na pisaniu oprogramowania dla projektu GNU, odczuwał nieznośny ból rąk i był zmuszony wynająć maszynistę. Dlatego teraz korzystał z klawiatury wymagającej o wiele słabszego naciskania na klawisze niż w przypadku klawiatur standardowych.

Podczas pracy wyłączał się na wszelkie bodźce zewnętrzne.

Patrząc na jego oczy skupione na ekranie i palce tańczące po klawiaturze, szybko dochodziło się do wniosku, że tak wygląda współpraca dwóch dobrych przyjaciół .

Zakończył pisanie kilkoma głośniejszymi uderzeniami klawiszy i zamknął laptopa.

"Gotów na lunch?" spytał .

Poszliśmy do mojego samochodu. Tłumacząc się bólem w kostce, siedł wolno, nieco utykając. Twierdził , że to wynik uszkodzenia ścięgna lewej stopy. Kontuzja miała miejsce trzy lata temu i była na tyle poważna, że on, zaprzysięgły fan tańców ludowych, był zmuszony do rezygnacji z nich. "Miłość do tańca to moja cecha wrodzona" - narzekał . - "Niemożność tańczenia to dla mnie tragedia".

Jego ciało było obrazem tej tragedii. Brak ruchu spowodował nabrzmienie policzków i pojawienie się brzuszka, który przed rokiem był niemal niewidoczny. Widać było, że przybranie na

wadze było dla niego dramatem, gdyż idąc odchyłał się do tyłu jak kobieta w ciąży starająca się zrównoważyć nienormalny ciężar.

Dodatkowo spacer stał się jeszcze wolniejszy, gdyż Stallman zatrzymywał się, by wąchać róże. Spotykając wyjątkowo piękny kwiat, łaskotał wewnętrzne płatki swym okazałym nosem, wciągał zapach i odchyłał się z wyrazem ukontentowania na twarzy.

"Mmm, rhinophytophilia9" - powiedział, pocierając kark. Jazda samochodem do restauracji nie trwała dłużej niż trzy minuty. Zgodnie z zaleceniem Tima Neya, niegdyś dyrektora Fundacji Wolnego Oprogramowania, pozostawił em Stallmanowi wybór restauracji. Niektórych dziennikarzy zmylił niemal ascetyczny styl życia Stallmana - we wszystkim, co dotyczy jedzenia, jest on hedonistą. Dodatkową korzyścią płynącą z bycia wędrownym misjonarzem głoszącym ideę wolnego oprogramowania jest możliwość próbowania kuchni różnych narodów. "Kiedy Richard przyjeżdża do dowolnego dużego miasta na świecie, z dużą dozą prawdopodobieństwa można założyć, że zna tam najlepsze restauracje" - twierdzi Ney. "Jest również bardzo dumny ze znajomości potraw w menu i z wielką radością składa zamówienia dla wszystkich przy stole."

Na dzisiejszy posiłek wybrał restaurację dim sum10 w stylu kantońskim, dwie przecznice od University Avenue, głównej ulicy Palo Alto. Wybór był częściowo zainspirowany jego niedawna wizytą w Chinach i wygłoszeniem wykładu w prowincji Guangdong oraz awersją do ostrych przypraw charakteryzujących kuchnię Seczuanu i prowincji Hunan. "Nie jestem zwolennikiem ostrych przypraw" - przyznał.

Przyjechaliśmy kilka minut po 11 przed południem i już musieliśmy poczekać około 20 minut. Znajac hakerską niechęć do tracenia czasu, obawiałem się wybuchu, ale wbrew oczekiwaniom Stallman spokojnie przyjął to do wiadomości.

"Szkoda, że nie ma nikogo, kto mógłby się do nas przyłączyć" - powiedział. - "Zawsze weselej jeść w większej grupie".

Czekając, ćwiczył kilka kroków tanecznych. Jego ruchy były niepewne lecz poprawne. Dyskutowaliśmy o sprawach bieżących. Powiedział, że żałuje jedynie, iż na konferencji Linux-World nie mógł uczestniczyć w powołaniu Fundacji GNOME. Fundacja wspierana przez Sun Microsystems i IBM jest w znacznej mierze potwierdzeniem słuszności jego twierdzeń, że wolny rynek i wolne oprogramowanie nie muszą się wzajemnie wykluczać. Był niezadowolony, że ominęła go okazja do wypowiedzenia się.

"Nie podoba mi się sposób prezentacji. Obie firmy mówił ydużo o Linuksie, nawet nie wspominając o projekcie GNU" - mówił niezadowolony.

To niedocenianie kontrastuje z coraz lepszymi informacjami napływającymi z innych kontynentów, zwłaszcza z Azji - podkreśla. Krótkie spojrzenie na plan jego podróży w roku 2000 świadczy o niewątpliwym wzroście popularności wolnego oprogramowania na świecie. Pomiedzy ostatnimi wizytami w Indiach, Chinach i Brazylii na 115 dni tylko 12 spędził w USA. Podróże dają mu możliwość sprawdzenia, w jaki sposób koncepcja wolnego oprogramowania jest przyjmowana przez ludzi różnych języków i kultur.

"W Indiach wiele osób interesuje się wolnym oprogramowaniem, gdyż widzą w nim możliwość budowania własnej infrastruktury informatycznej bez wydawania zbyt wielkich sum pieniędzy" - stwierdził Stallman. - "W Chinach idea wolnego oprogramowanie szerzy się wolniej, co jest związane z brakiem wolności słowa. Trudno przyjmować wolne oprogramowanie tam, gdzie nie ma wolności wypowiedzi.

Pomimo to stopień zainteresowania wolnym oprogramowaniem jest znaczny, co zaobserwowałem podczas mojej wizyty".

Rozmowa zeszła na Napstera, firmę z San Mateo w Kalifornii, która w ciągu kilku ostatnich miesięcy stała się przedmiotem wielkiego zainteresowania mediów. Dostarcza ona narzędzi pozwalających jednym fanom muzyki przeglądać i kopiować pliki muzyczne należące do innych. Dzięki rosnącej roli

internetu ten tzw. program peer-to-peer¹¹ rozrósł się i przekształcił w prawdziwą szafę grającą, dając zwykłym fanom muzyki sposób na słuchanie ulubionych utworów z odgrywanych na komputerze plików MP3 bez płacenia złamanego grosza - ku rozgoryczeniu firm fonograficznych. System Napstera, choć korzysta z oprogramowania z prywatnymi prawami własności, czerpie inspirację z głoszonej przez Stallmana idei, że wszystko, co staje się cyfrową rzeczywistością, w której kopiowanie informacji staje się o wiele ważniejsze od kopiowania dźwięków i materii, coraz trudniej podlega jakimkolwiek restrykcjom. Zamiast nakładać kolejne ograniczenia, rządzący Napsterem zdecydowali się czerpać korzyści z naturalnego pędu i tworząc dla słuchaczy muzyki centralny rynek handlu i wymiany plików, postawili na sterowanie ich zainteresowaniami w stronę innych możliwości handlowych.

Nagły sukces modelu zaproponowanego przez Napstera przeraził tradycyjne firmy fonograficzne. Na kilka dni przed moim spotkaniem ze Stallmanem w Palo Alto Marilyn Patel, U. S. District Court Judge¹², pozytywnie rozpatrzył złożone przez Recording Industry Association of America¹³ żądanie wydania sądowego zakazu działania systemu wymiany plików. Zakaz ten został z kolei zawieszony przez U. S. Ninth District Court of Appeals¹⁴. Jednakże na początku 2001 roku również sąd apelacyjny uznał, że firma z San Mateo złamała prawo autorskie, co rzecznik RIAA, Hillary Rosen, ogłosił jako "wielkie zwycięstwo twórczej społeczności i legalnego rynku internetowego¹⁵".

Dla hakerów takich jak Stallman model biznesowy Napstera jest nie do przyjęcia z kilku powodów. Firma gorliwie korzysta z hakerskich zasad, takich jak otwarte udostępnianie plików i społeczna własność informacji, jednocześnie sprzedając usługi oparte na korzystaniu z oprogramowania z prywatnym prawem własności, co tworzy przedziwną mieszaninę i przekazuje społeczeństwu niejasne przesłanie. Jako osoba, która masę

czasu spędziła na ostrożnym formułowaniu informacji przeznaczonych do upublicznienia, Stallman ostrożnie i oszczędnie wyraża swe sądy o Napsterze. Przyznaje, że nauczył się kilku rzeczy dotyczących społecznej strony fenomenu Napstera.

Mówi: "Przed sprawą Napstera twierdziłem, że jest OK, jeżeli ludzie zajmują się prywatną redystrybucją plików służących rozrywce. Jednakże liczba ludzi, która korzysta z serwisu Napstera, uświadomiła mi, że redystrybucja kopii na wielką skalę jest sprawą o wielkim znaczeniu i nie można jej lekceważyć".

Gdy Stallman to powiedział, drzwi restauracji otworzyły się i właściciel zaprosił nas do środka. W ciągu kilku sekund zostaliśmy posadzeni w narożniku sali obok wielkiej zwierciadlanej ściany. Menu restauracji było jednocześnie formularzem zamówienia i zanim kelner zdążył przynieść wodę, Stallman szybko zakreślił odpowiednie kwadraciki. "Krewetki smażone na głębokim oleju w cieście z sera sojowego" - przeczytał. - "Ser sojowy ma taką interesującą fakturę. Myślę, że powinniśmy to zamówić".

Ten komentarz był wstępem do podjętej in promptu rozmowy o kuchni chińskiej i niedawnej wizycie Stallmana w Chinach. "Jedzenie w Chinach jest wspaniałe" - rzekł głosem, w którym po raz pierwszy tego dnia wyczułem emocje. "Tyle różnych rzeczy, których nigdy nie widziałem w USA, miejscowe potrawy z lokalnych grzybów i lokalnych warzyw. Zaczął on nawet prowadzić dziennik posiłków, tylko po to, by nie tracić informacji o cudownościach, które mi podawano".

Rozmowa przeszła na zagadnienia kuchni koreańskiej. Podczas tej samej azjatyckiej podróży w czerwcu 2000 r. Stallman odwiedził także Koreę Południową. Jego przybycie wywołało miniburzę w lokalnych mediach, a to z powodu ogólnokoreańskiej konferencji oprogramowania, w której w tym czasie brał udział założyciel i prezes zarządu Microsoftu, Bill Gates. Poza ujrzeniem swego zdjęcia bezpośrednio nad zdjęciem

Billa Gatesa na pierwszej stronie seulskiej gazety, najlepszym wspomnieniem Stallmana z Korei był a tantejsza kuchnia.

"Dostał ę miskę naeng myun, czyli makaronu ryżowego. W różnych miejscach do przygotowania naeng myun używa się różnego makaronu ryżowego. Mogę z całą pewnością stwierdzić, że dostał ę wówczas najwspanialsze naeng myun, jakie kiedykolwiek jadłem".

Stallman często używa określenia "wspaniał y". Byłem o tym przekonany po kilku minutach słuchania jego hymnu pochwalnego na temat naeng myun. Cały czas czułem jego laserowy wzrok wypalający znamię na moim prawym ramieniu.

"Co za wspaniał a kobieta siedzi za tobą" - usłyszał ę.

Obróciłem się, zerkając na plecy siedzącej tam niewiasty. Była młoda, około 25 lat, ubrana w białą sukienkę z cekinami. Ona i mężczyzna, który jej towarzyszył, właśnie płacili rachunek. Gdy wstali od stołu, by wyjść, nie musiał ę nawet patrzeć, gdyż oczy Stallmana wprost przyćmił a intensywność przeżycia.

"O nie" - powiedział . - "Wyszli i pewnie już nigdy jej nie zobaczę".

Ale po krótkim westchnieniu powrócił do równowagi. Ta chwila dała mi szansę na skierowanie rozmowy na jego reputację osobnika będącego zaprzeczeniem uczciwego seksu. Była to nieco dyskusyjna opinia. Wielu hakerów wspominał o jego skłonności do całowania kobiet w rękę przy powitaniu¹⁶.

Salon.com w artykule z 26 maja sportretował Stallmana jako hakerskiego uwodziciela. Dokumentując związek wolnego oprogramowania z wolnym seksem, reporterka Annalee Newitz pokazała Stallmana jako przeciwnika wszelkich tradycyjnych wartości rodzinnych i zwolennika miłości, ale nie monogamii¹⁷.

Gdy poruszyłem ten temat, Stallman nieco odsunął menu.

"Tak, większość mężczyzn wydaje się pożądać seksu i wydaje się mieć nieco pogardliwy stosunek do kobiet" - powiedział - "a kobiety zdają się to aprobować, czego zupełnie nie mogę zrozumieć".

Przypomni¹⁸ł em fragment z wydanej w roku 1999 książki Open Sources, gdzie przyznał się, że feralne jądro GNU chciał nazwać imieniem swojej ówczesnej przyjaciółki. Imię brzmiało Alix i doskonale pasowało do przyjętej w środowisku Uniksa konwencji używania x na końcu nazwy nowego jądra, czego przykładem jest Linux. Ponieważ kobieta była administratorką Uniksa, byłoby to dla niej tym bardziej wzruszające. Niestety, w końcu główny twórca tego jądra nadał mu nazwę HURD¹⁸. Choć potem Stallman rozstał się z tą przyjaciółką, historia ta rodzi pytanie, czy zgodnie z opisem mediów jest fanatykiem o dzikim spojrzeniu, czy beznadziejnym romantykiem, wędrującym Don Kichotem walczącym z wiatrakami wielkich korporacji w nadziei zrobienia wrażenia na jeszcze nieokreślonej Dulcynei?

"Nie próbowałem być romantyczny" - mówi, wspominając historię Alix. - "Był a to raczej chęć podroczenia się. Było to romantyczne, lecz również nieco zł¹⁹ośliwe. Rozumiesz? Mogłoby to być zachwycająca niespodzianka".

Uśmiechnął się po raz pierwszy tego przedpołudnia. Podniosłem rękę, całując ją. "Tak, robię to" - potwierdził. - "Uznałem, że jest to sposób wyrażenia uczucia akceptowany przez wiele kobiet. Daje to szansę pokazania uczucia i uzyskania jego akceptacji". Uczucie jest nicią przewijającą się przez całe życie Richarda Stallmana i pytany o to jest szczery aż do bólu. "W przeciwieństwie do myśli, w moim rzeczywistości życiu nie było wiele uczucia" - mówi. Rozmowa staje się niezręczna. Po kilku następnych słowach Stallman znów podnosi do oczu menu, kończąc tym ten wątek.

"A co powiesz na nieco shimai?" - pyta.

Gdy przyniesiono potrawy, rozmowa zaczęła przeskakiwać z tematu na temat. Rozmawialiśmy o częstym wśród hakerów umiłowaniu chińskiej kuchni, cotygodniowych obiadach w bostońskiej chińskiej dzielnicy w czasach, gdy Stallman był programistą w AI Lab, oraz o wewnętrznej logice języka chińskiego i wynikającym z tego systemie zapisu. Każde moje

stwierdzenie spotykał się z błyskotliwą odpowiedzią Stallmana.

"Podczas ostatniego pobytu w Chinach słyszałem kilka osób mówiących w dialekcie szanghajskim." - powiedział. - Warto było oposłuchać. Brzmiało to całkiem odmiennie (od mandaryńskiego). Musiałem ich poprosić, aby powiedzieli kilka słów w obu odmianach języka. Czasami widać podobieństwo, a interesowało mnie, czy dźwięki są podobne. Zwykle nie są. Interesuje mnie to, gdyż istnieje teoria, że wszystkie wmawiane słowa ewoluowały przez utratę dodatkowych sylab, gubionych i zastępowanych. Efektem jest brzmienie języka. Jeżeli to prawda, a widziałem prace, w których dowodzono, że proces ten odbywał się w czasach już historycznych, dialekty przed procesem tracenia sylab powinny mieć jeden wspólny rdzeń. Podano pierwsze danie, talerz smażonych na patelni naleśników z rzepy. Zamilkliśmy na chwilę, aby wykroić sobie po sporym prostokątnym kawałku, którego zapach przypominał gotowaną kapustę, lecz smak raczej placki ziemniaczane podsmażone na bekonie.

Postanowiłem powrócić do rozmowy na zasadnicze tematy. Byłem ciekaw, czy jego życie nastolatka przygotowało go do podejmowania kontrowersyjnych poczynań. Myślałem głównie o jego trwającej od roku 1994 batalii o zastąpienie nazwy Linux przez GNU/Linux.

"Wierzę, że to mi pomogło" - odparł, przeżuając. - "Nigdy nie rozumiałem, co oznacza presja wywierana przez rówieśników. Myślę, że byłem na tyle odrzucony przez innych, że nie spodziewałem się niczego zyskać, postępując zgodnie z aktualną modą. Nic by mi to nie dało. Nadal pozostałbym tak samo na uboczu".

Wskazał na swój gust muzyczny jako odbicie tendencji do pozostawania w opozycji. Gdy większość kolegów z ogólniaka słuchała Motown i rocka, on wołał muzykę klasyczną. Pamięta, że w szkole średniej prowadziło to do zabawnych zdarzeń. W roku 1964 po występie Beatlesów w programie "Ed Sullivan

Show" 19 większość kolegów ruszył do sklepów, wykupując ostatni album i single tego zespołu. Stallman zaś postanowił zbojkotować słynną czwórkę.

"Lubiłem część przedbeatlesowej muzyki popularnej" - mówi Stallman, - "ale nie lubiłem Beatlesów. Zwłaszcza nie podobała mi się dzika reakcja ludzi na ich koncerty. Wszyscy zabiegali o ich występy, aby się przed nimi płaszczyć".

Gdy bojkot Beatlesów pozostał niezauważony, poszukał innych metod, by wytknąć rówieśnikom mentalność stada. Przez chwilę zastanawiał się nad założeniem zespołu rockowego, którego zadaniem byłoby parodiowanie grupy z Liverpoolu.

"Chciałem ten zespół nazwać Tokyo Rose and Japanese Beetles20".

Ponieważ wiadomo, że lubi muzykę ludową różnych krajów, spytałem, czy równie podoba mu się Bob Dylan i inni wykonawcy muzyki ludowej z wczesnych lat sześćdziesiątych. Potrząsnął głową i powiedział :- "Lubiłem Petera, Paula i Mary. Przypominali mi wielki filk".

Gdy spytałem, co znaczy "filk", wyjaśnił, że to popularna pieśń lub piosenka, w której liryka została zastąpiona przez parodię liryki. Taki proces przerabiania to "filking" i jest to popularne zajęcie hakerów i miłośników science fiction. Przykłady to "On Top of Spaghetti" - przeróbka "On Top of Old Smokey" oraz filkowy majstersztyk "Weirda" Ala Yankovica21 - "Yoda", wykonywana w stylu "Gwiezdnych wojen" przeróbka "Loli" zespołu The Kinks.

Zapytał mnie, czy mam ochotę na wystąpienie ludowego filku. Gdy tylko zdążyłem odpowiedzieć twierdząco, zaczął śpiewać niespodziewanie czystym głosem:

How much wood a woodchuck chuck,
If a woodchuck could chuck wood?
How many poles could a polak lock,
If a polak could lock poles?
How many knees could a negro grow,
If a negro could grow knees?

The answer, my dear, is stick it in your ear.

The answer is to stick it in your ear.

Śpiew dobiegł końca i na ustach Stallmana pojawił się kolejny niemal dziecinny półuśmiech. Rozejrzał się po stolikach dookoła. Azjatycka rodzina ciesząca się niedzielnym lunchem prawie nie zwrócił uwagi na brodacza w średnim wieku śpiewającego altem²². Po chwili wahania roześmiałem się również.

"Czy masz ochotę na ostatniego cornballa²³?" - spytał Stallman i zanim zdążyłem odpowiedzieć, schwycił pałeczkami inkrustowaną ziarnem kulkę. Unosząc ją dumnie, powiedział: "Jednak to ja będę tym, który to dostanie".

Skończyliśmy jedzenie i rozmowa przybrała formę normalnego wywiadu. Stallman usadowił się w pozycji półleżącej z filiżanką herbaty w dłoniach. Wróciliśmy do sprawy Napstera i jej związków z ruchem wolnego oprogramowania. Spytał mnie, czy zasady wolnego oprogramowania powinny być rozciągnięte na podobne obszary, na przykład utwory muzyczne.

"Przenoszenie rozwiązań z jednej dziedziny na inną jest błędem" - powiedział Stallman, odróżniając nagrania muzyczne od oprogramowania. - "Na każdy rodzaj pracy trzeba patrzeć inaczej i wyciągać odpowiednie wnioski".

Mówiąc o prawach autorskich, wyróżnił trzy kategorie. Pierwsza to produkty "funkcjonalne", np. oprogramowanie, słowniki i książki. Druga obejmuje prace, które można uznać za oświadczenia lub opisy zdarzeń, np. prace i dokumenty historyczne. Wszelkie zmiany wprowadzane przez kolejnych czytelników mogłyby całkowicie zniweczyć cel tych prac i sens ich tworzenia. Wreszcie ostatnia kategoria to dzieła osobiste, np. dzienniki, wspomnienia i autobiografie. Modyfikowanie takich dokumentów oznaczałoby zmianę osobistych wspomnień lub punktów widzenia autorów, co Stallman uznał za etycznie niedopuszczalne.

Tylko pierwsza z trzech wyliczonych kategorii daje użytkownikowi nieograniczone prawo wprowadzania zmian, zaś

w kategorii drugiej i trzeciej zmiany mogą być wprowadzane jedynie za zgodą autora oryginału. Jednakże, twierdził uparcie, wolność kopiowania i niekomercyjnej redystrybucji powinna pozostać nienaruszona w przypadku wszystkich trzech wymienionych kategorii. Oznacza to danie użytkownikom internetu prawa do generowania setek kopii artykułów, obrazów, piosenek lub książek i rozsyłania ich pocztą elektroniczną. "Jest oczywiste, że prywatna, okazjonalna redystrybucja nie powinna być zabroniona, gdyż zakazywana jest jedynie w państwach policyjnych. Wtrącanie się między znajomych i przyjaciół to działanie aspołeczne. Napster przekonał mnie, że potrzebne jest także zezwolenie na niekomercyjną redystrybucję rozrywki. Po prostu dlatego, że jest to potrzebne tak wielu ludziom i tak wielu ludziom przynosi to pożytek".

Gdy spytałem, czy sądy podziela tak permissywną opinię, przerwał mi w pół zdania.

"To źle postawione pytanie" - powiedział. - "Teraz zmieniłeś przedmiot rozważań, przechodząc od zagadnień etycznych do sposobu interpretacji przepisów prawa. A to są dwa zupełnie różne spojrzenia na tę samą kwestię. Bez sensu jest skakać z jednego pola na drugie. Sądy zwykle surowo interpretują istniejące przepisy prawa, gdyż taka interpretacja jest kupiona przez wydawców".

Ten komentarz pozwala zajrzeć w głębię politycznej filozofii Stallmana: to, że legalna władza wspiera skłonność biznesu do traktowania praw autorskich oprogramowania tak samo jak prawa własności gruntu, nie oznacza, że użytkownicy komputerów mają się temu podporządkować. Wolność to sprawa etyki, a nie prawa. "Patrząc nie na to jakie jest istniejące prawo, ale jakie powinno być. Staram się naszkicować jego zarys. Myślę, jak prawo powinno działać. Uważam, że prawo zabraniające udostępnienia przyjacielowi kopii jest moralnym odpowiednikiem prawa Jima Crowa²⁴. Trudno takie prawo darzyć szacunkiem".

Przywołanie Jima Crowa rodzi kolejne pytanie. Jak dalece na Stallmana mają wpływ liderzy polityczni z przeszłości? Na podobieństwo ruchu praw obywatelskich z lat pięćdziesiątych i sześćdziesiątych również jego wysiłki kierowania zmianami społecznymi pozostają pod wpływem zawsze atrakcyjnych haseł wolności, sprawiedliwości i postępowania fair play.

Stallman słuchał, dzieląc uwagę między wypowiedianą przeze mnie analogię i szczególnie splątany kosmyk włosów. Gdy dotarł do porównania go do doktora Martina Luthera Kinga, przygryzł koniec kosmyka i przerwał mój wywód.

"Nie jestem z tej drużyny, ale gram w tę samą grę" - powiedział, żując.

Zaproponowałem Malcoma X25 jako inny punkt odniesienia. Analogicznie do byłego rzecznika Nation of Islam²⁶ Stallman zdobył sobie opinię osoby kontrowersyjnej, odgradzającej się od potencjalnych sojuszników i wynoszącej samowystarczalność ponad integrację kulturową.

Przeżuwając koniec innego kosmyka długich włosów, odrzucił i to porównanie. "Moje przesłanie jest bliższe temu, co głosił Martin Luther King" - powiedział. "Jest to przesłanie uniwersalne głoszące potępienie pewnych praktyk znęcania się nad innymi. Nie głoszę nienawiści do nikogo i nie zwracam się do żadnej wybranej, wąskiej grupy. Wzywam wszystkich do docenienia wolności i jej zachowania".

Jednakże skłonność do podejrzliwego traktowania wszelkich politycznych aliansów jest fundamentalną cechą jego charakteru. Można zrozumieć jego niechęć do terminu "otwarty kod źródłowy" (ang. open source) i do udziału w ostatnich projektach koalicyjnych. Przez dwie dekady wbijał ludziom do głowy ideę wolnego oprogramowania i to określenie jest jego kapitałem politycznym. Niemniej jego komentarze w rodzaju dowcipu o Hanie Solo na LinuxWorld w roku 1999 jedynie umacniają w przemyśle oprogramowania jego opinię rozczarowanego odludka odcinającego się od politycznych i gospodarczych trendów.

"Lubię i szanuję Richarda za to, czego dokonał" - mówi Robert Young, prezes zarządu Red Hat, - "ale mam do niego żal za to, że często przyjaciół traktuje gorzej od wrogów".

Jego niechęć do szukania sojuszników jest równie niezrozumiała, gdy rozważamy dziedziny zainteresowań politycznych niezwiązanych z ruchem wolnego oprogramowania. Krótka wizyta w jego biurze w MIT przywodzi na myśl biuro wymiany artykułów prasowych i komunikatów agencyjnych na temat gwałcenia praw ludzkich jak świat długi i szeroki. Na jego stronie internetowej znajdziemy diatriby na temat Digital Millennium Copyright Act²⁷, wojny narkotykowej i Światowej Organizacji Handlu.

Widząc jego skłonności działacza politycznego, spytał em, dlaczego nie stara się mówić donośniejszym głosem. Dlaczego swej pozycji w świecie hackerskim nie stara się wykorzystać jako platformy do rozszerzania swych wpływów, zamiast usuwać się w cień.

Pozwolił opaść zmierzwionemu kosmykowi włosów i przed udzieleniem odpowiedzi zamyślił się na chwilę.

"Obawiam się, by nie przesadzać z ważnością tego bajorka wolności" - powiada. - "Jest tak wiele dobrze znanych tradycyjnych obszarów walki o wolność i lepsze społeczeństwo i są one tak ważne, że nie mogę głosić, że sprawa wolnego oprogramowania jest równie ważna. To odpowiedzialność, która na mnie spadła i której staram się podołać. Sądzę, że mogę tu coś zrobić. Ale na przykład zwalczanie brutalności policji, zakończenie wojny narkotykowej, zniszczenie rasizmu i zapewnienie każdemu godziwego, wygodnego życia, zapewnienie prawa do aborcji, ochronienie nas przed teokracją - to działania strasznie ważne, o wiele ważniejsze od tego, co ja robię. Chciałbym wiedzieć, w jaki sposób mogę je wesprzeć".

Po raz kolejny Stallman pokazał, że jego działalność polityczna wypływa z głębokiego osobistego przekonania. Wiele czasu zabrało mu stworzenie i rozwinięcie ruchu wolnego oprogramowania oraz dopracowanie szczegółów jego

zasadniczych reguł. Dlatego waha się przed udziałem w każdym ruchu, który mógł by go wynieść poza znane granice na ziemię, której nie ma na żadnej mapie.

"Chciałbym wiedzieć, w jaki sposób można rozwiązać te wielkie problemy. Był bym z tego bardzo dumny, ale są to rzeczy tak trudne, że wielu lepszych ode mnie i pracujących nad tym od lat posunęło sprawy naprzód jedynie o kilka kroków lub wcale" - mówi. - "Ale gdy inni są zajęci wielkimi sprawami i walczą z wielkimi zagrożeniami, ja dostrzegłem jeszcze coś, co nam grozi, i występuję przeciwko tej groźbie. Być może nie jest to wielkie zagrożenie, ale ja jestem sam na polu walki".

Żując koniuszek kosmyka, zaproponował zapłacenie rachunku. Zanim kelner zdołał zainkasować zapłatę i odejść, Stallman wyciągnął biały banknot dolarowy i dorzucił go do kupki już leżących na stole. Z daleka było widać, że jest fałszywy. Nie mogłem się powstrzymać, by go nie obejrzeć. Zamiast portretu George'a Washingtona lub Abrahama Lincolna zobaczyłem rysunek świnki. Napis United States of America został zastąpiony przez United Swines of Avarice²⁸. Na banknocie podano wartość zero dolarów. Gdy kelner zbierał banknoty, Stallman pociągnął go za rękaw.

"Jako ekstranapiwek dodałem jedno zero" - oświadczył z półuśmiechem na ustach.

Ogłupiały i nic nierozumiejący kelner zabrał banknoty ze stołu i z uśmiechem pośpiesznie się oddalił.

"Myślę, że możemy już iść" - powiedział Stallman.

Przypisy

1 Święty wolnego oprogramowania - przyp. tłum.

2 Patrz: Andrew Leonard The saint of Free Software, Salon.com (sierpień 1998):

http://archive.salon.com/21st/feature/1998/08/cov_31feature.html.

3 Patrz: Leander Kahney Linux' sForgotten Man, "Wired

News" (5 marca 1999):

<http://www.wired.com/news/technology/0,1282,18291,00.html>

4 Patrz: "Programmer on moral high ground; Free software is a moral issue for Richard Stallman believes in freedom and free software". London Guardian (6 listopada 1999).

To tylko kilka przykładów religijnych porównań. Dotychczas najdalej idące porównanie możemy znaleźć u Linusa Torvaldsa, który w autobiografii (patrz: Linus Torvalds i David Diamond Just for Fun: The Story of an Accidental Revolutionary - HarperCollins Publishers, Inc., 2001, str. 58) pisze: "Richard Stallman jest bogiem wolnego oprogramowania".

Zaszczytną wzmiankę o Stallmanie znajdziemy u Larry'ego Lessiga, który w przypisie do swej książki The Future of ideas (Random House, 2001, str. 270) porównuje Stallmana do Mojżesza:

...był inny przywódca Linus Torvalds, który jak Mojżesz poprowadził ruch ku ziemi obiecanej przez ułatwienie rozwoju ostatniego elementu układanki systemu operacyjnego. Również Stallman jak niegdyś Mojżesz jest zarówno poważany i znieważany przez pobratymców z ruchu. Jest on nieprześląganym więc dla wielu również inspirującym przywódcą istotnie ważnego aspektu współczesnej kultury. Czuję głęboki respekt dla zasad i przekonań tej wyjątkowej indywidualności, jak również dla tych, którzy mają odwagę mu się przeciwstawiać, narażając się na jego gniew.

Podczas ostatniego wywiadu spytałem Stallmana, co sądzi o tych religijnych porównaniach. "Niektórzy rzeczywiście porównują mnie do starotestamentowych proroków, a to dlatego, że często ganili oni pewne społeczne praktyki. W kwestiach moralnych nie szli na żaden kompromis, nie można był oich przekupić i często traktowano ich ze wzgardą".

5 Połączona z targami konferencja świata linuksowego, czyli firm i osób zainteresowanych tym systemem. Informacje można

znaleźć pod adresem: <http://www.linuxworldexpo.com> - przyp. tłum.

6 Patrz: jeden z poprzednich przypisów: Leander Kahney Linux's Forgotten Man.

7 Nagroda Linusa Torvaldsa za służbę dla społeczeństwa - przyp. tłum.

8 Dotyczy postaci z Gwiezdných wojen - przyp. tłum.

9 Wówczas myślałem, że to naukowa nazwa kwiatu. Kilka miesięcy później dowiedziałem się, że rhinophytophilia to humorystyczne nazwanie tego, co robił, czyli wsadzania nosa w kwiat i napawania się jego zapachem. Inne humorystyczne informacje o Stallmanie i kwiatach można znaleźć pod adresem: <http://www.stallman.org/articles/texas.html>

10 Dim sum - rodzaj tradycyjnej kuchni chińskiej - przyp. tłum.

11 Równy-do-równego lub rówieśnik-do-rówieśnika - przyp. tłum.

12 Sędzia sądu dystryktowego - przyp. tłum.

13 Amerykańskie stowarzyszenie przemysłu fonograficznego - przyp. tłum.

14 Sąd apelacyjny 9. dystryktu - przyp. tłum.

15 Patrz: "A Clear Victory for Recording Industry in Napster Case", komunikat prasowy RIAA (12 lutego 2001): http://www.riaa.com/PR_story.cfm?id=372

Powyższy adres WWW jest nieaktualny, przynajmniej w chwili tłumaczenia tego rozdziału książki. Z komunikatami prasowymi RIAA dotyczącymi sprawy Napstera można się zapoznać pod adresem: <http://www.riaa.com/news/filings/napster.asp> - przyp. tłum.

16 Patrz: Mae Ling Mak "Mae Ling Story" (17 grudnia 1998): <http://www.crackmonkey.org/pipermail/crackmonkey/1998q4/003006.html>.

Na razie Mak jest jedyną kobietą, która jest skłonna mówić i pisać o tym, choć ustną relację słyszał emod kilku innych. Mak pomimo wcześniejszej niechęci, potem odłożył a na bok

uprzedzenia i tańczył aże Stallmanem na LinuxWorld w 1999 r.

17 Patrz: Annale Newitz "If Code is Free Why Not Me?", Salon.com (26 maja 2000):

http://archive.salon.com/tech/feature/2000/05/26/free_love/print.html.

18 Patrz: Richard Stallman "The GNU Operating System and Free Software Movement", Open Sources (O' Rilly & Associates, Inc., 1999, str. 65).

19 Patrz: <http://www.hp.com.pl/doc/315> - przyp. tłum.

20 Tokijska róża i japońskie chrząszcze. Ponieważ nazwa zespołu The Beatles pochodzi od zniekształconej pisowni słowa "beetles" ("chrząszcze", "żuki", potocznie "robaki"), można snuć rozważania na temat podtekstów ukrytych w nazwie wymyślonej przez Stallmana - przyp. tłum.

21 "Weird" Al Yankovic lub "Weird Al" Yankovic - "weird" to "dziwaczny", ale podanie tłumaczenia "dziwaczego Ala Yankovica" nie oddaje atmosfery i stylu tego połączenia pseudonimu z imieniem i nazwiskiem. Ciekawych odsyłam pod adres internetowy: <http://www.weirdal.com/> - przyp. tłum.

22 Więcej filków Stallmana znajdziesz na stronie:

<http://www.stallman.org/doggerel.html>. Aby posłuchać Stallmana śpiewającego "The Free Software Song", odwiedź stronę <http://www.gnu.org/music/free-software-song.html>

23 Tutaj kulka z popcornów zlepionych melasą. Ciekawostką jest znaczenie slangowe, które podaję według internetowego suplementu do "Wielkiego słownika angielsko-polskiego PWN-Oxford": <http://www.serwistlumacza.com/texts/poprawki.html>

"cornball - raczej pierdoł niż pocziwiec, to drugie określenie trudno bowiem uznać za pejoratywne" - przyp. tłum.

24 Prawa Jima Crowa - ustawodawstwo legalizujące segregację rasową w USA; nazwa od pogardliwego przezwiska Murzyna - Jim Crow. Więcej informacji pod adresem:

http://encyklopedia.pwn.pl/32398_1.html - przyp. tł um.

25 Malcolm X (wł. ąściwie Malcolm Little) - urodzony w roku 1926 przywódca nacjonalistycznego ruchu Murzynów w USA. Podczas odsiadki kary za włamanie przył ączył się do więziennej sekty czarnych muzułmanów. Po wyjściu na wolność prowadził walkę z separatyzmem rasowym. Uznawał przemoc za usprawiedliwioną. Potem nieoczekiwanie zmienił poglądy i założył w roku 1964 Islamską Socjalistyczną Organizację Jedności, propagującą hasła solidarności rasowej. Rok później podczas przemówienia w Harlemie zginął prawdopodobnie z rąk członków Narodu Islamu - patrz następny przypis. W roku 1964 wydano jego książkę Autobiography of Malcolm X - przyp. tł um.

26 Naród Islamu - murzyńska organizacja religijno-polityczna przez niektórych nazywana sektą, założona w roku 1931. Atakuje białych, głównie zaś Żydów, obarczając ich odpowiedzialnością za nędzę i upokorzenia, jakich doznaje znaczna część społeczności murzyńskiej. Jednym z jej aktywnych dział ący był Malcolm X. Wystąpił z niej po zmianie poglądów w roku 1964 - patrz poprzedni przypis. Więcej informacji można znaleźć w Internecie np. pod adresami: <http://www.planetaislam.com/innowacje/nationofislam.html> lub <http://www.niniwa2.cad.pl/black.htm> - przyp. tł um.

27 W skrócie DMCA (Ustawa o prawie autorskim w środowisku cyfrowym)

- informacja o jej wpł ywie na ruch wolnego oprogramowania można przeczytać po polsku na stronie

<http://www.gnu.org/philosophy/words-to-avoid.pl.html>.

Przytoczył em wtekście nazwę w wersji angielskiej, gdyż zwykle tak jest podawana nawet w polskiej prasie (w przeciwieństwie np. do Światowej organizacji Handlu) - przyp. tł um.

28 Gra słów nie do oddania po polsku; United States of America to Stany Zjednoczone Ameryki, a United Swines of Avarice dosłownie należał dby przełożyć jako "zjednoczone świnie chciwości", co po polsku brzmi dziwnie - przyp.

tl um.

6. WSPÓLNOTA EMACS

W latach siedemdziesiątych AI Lab było miejscem ze wszech miar szczególnym. Specjaliści najwyższej klasy, pracujący nad projektami dotyczącymi najnowszych technologii, zapewniali mu znaczącą pozycję w świecie nauk komputerowych.

Jednakże wewnętrzna kultura hackerska i towarzysząca jej anarchia prowadziły do buntowniczego mistycyzmu. Później, gdy większość naukowców i gwiazd programowania odeszło gdzie indziej, hackerzy mogli swobodnie urzeczywistnić efemeryczny świat, który wcześniej zasiedlili.

W artykule opublikowanym w "Forbes" w roku 1998 przytoczono słowa Stallmana wspominającego laboratorium z tego okresu i panujący tam etos wzajemnego udostępniania sobie oprogramowania. Jak wspomina Stallman -

"Przypominało to ogród rajski. Nawet nie przychodziło nam do głowy, że można ze sobą nie współpracować"¹.

To biblijne porównanie, choć może przesadzone, podkreśla bardzo istotny fakt. Dla wielu dziewięciopiętrowy budynek przy Tech Square 545 był czymś więcej niż tylko miejscem pracy. Dla hackerów w rodzaju Stallmana był domem.

Słowo "dom" ma swoje szczególne miejsce w słowniku Stallmana. Z uwagi na krytyczny stosunek do rodziców Stallman wzbrania się nazwać domem jakiegokolwiek miejsce pobytu aż do zamieszkania w harwardzkim internacie Currier House. Opuszczenie go opisuje w sposób tragikomiczny. Z lat spędzonych w Harvardzie żałuje jedynie wygnania z akademika. Dopiero pytanie o powód wcześniejszego opuszczenia internatu uświadamia mi, że otrzymam klasyczną odpowiedź w stallmanowskim stylu.

"W Harvardzie prowadzili taką politykę, by pozbywać się tych, którzy zaliczyli zbyt wiele kursów".

Nie mając miejsca w internacie i chęci do powrotu do Nowego

Jorku, postąpił tak jak przed nim wielu innych hakerów, między innymi Greenblatt, Gosper i Sussman. Przyjęty do MIT na studia magisterskie², wynajął mieszkanie w pobliżu Cambridge, by wkrótce przekonać się, że AI Lab jest jego faktycznym domem. Przemawiając w roku 1986, następująco opisał czas spędzony tamże:

Mogłem pracować nieco więcej od innych, gdyż każdego roku z takich lub innych powodów przez kilka miesięcy nie miał em mieszkania i po kilka miesięcy spędzał em w laboratorium. Było tam zawsze bardzo wygodnie, a latem na dodatek chłodno. Ale nie był oto nic niezwykłego i często można się było tam natknąć na śpiących, gdzie popadło. Siedziało się tam, dopóki można był o hakować, i nie chcąc przerywać zajęcia, trwał o przy komputerze aż do zupeł nego opadnięcia z sił .Wtedy szukało się dowolnej płaskiej i miękkiej powierzchni, by się położyć i usnąć. Atmosfera był a całkiem nieformalna³.

Czasami domowa atmosfera w laboratorium sprawiał a problemy. To, co dla jednych był o wygodnym miejscem do przespania się, inni uważali za norę przepelnioną dymem "elektronicznego opium". W roku 1976 w książce Computer Power and Human Reason⁴ naukowiec z MIT, Joseph Weizenbaum, skrytykował komputerowych ł apserdaków-pomyeńców (ang. computer bums⁵). W ten sposób Weizenbaum nazwał hakerów zasiedlających komputerowe pokoje w AI Lab. Pisał : "Ich zmięte ubrania, niegolone twarze oraz niemyte i nieczesane włosy świadczą o zatraceniu świadomości istnienia wł asnych ciał i świata, w którym te ciała funkcjonują. Istnieją tylko po to, by korzystać z komputerów i komputerom służyć"⁶.

Niemal ćwierć wieku po opublikowaniu tej opinii Stallman nadal jeży się na jej wspomnienie. Podejmując polemikę, używa czasu teraźniejszego, jak gdyby Weizenbaum stał w pobliżu, w tym samym pokoju. "On chce, by ludzie byli tylko zawodowcami, pracującymi wyłącznie dla pieniędzy i zapominającymi o tym, co robią, tak szybko, jak to tylko

możliwe - mówi. - To, co on uważa za normalny bieg rzeczy, ja uważam za tragedię".

Jednakże życie hakera nie było wolne od takich tragedii.

Stallman ocenia swoje przeistoczenie się z weekendowego hakera w naturalizowanego "mieszkańca" AI Lab jako zbieg bolesnych niepowodzeń, których przetrwanie ułatwiał a mu jedynie euforia działalności hakerskiej. Pierwszym niepowodzeniem było ukończenie studiów na Harvardzie. Chcąc dalej studiować fizykę, zapisał się na studia magisterskie na MIT. Wybór uczelni był oczywisty i naturalny. Dawał mu szansę pójścia w ślady wielkich poprzedników, absolwentów tej uczelni: Wiliama Shockeya (1936)⁷, Richarda P. Feynmanna (1939)⁸ i Murraya Gel-Manna (1951)⁹ oraz o dwie mile skracał odległość do AI Lab i nowego komputera PDP-10. "Moje zainteresowania coraz bardziej biegły yku programowaniu, ale przez pewien czas myślałem, że dam radę zajmować się i jednym, i drugim" - wspomina.

Trudząc się nad zdobyciem stopnia naukowego w dzień, a w nocy programując w AI Lab, pracował jak mnich, który przez narzucenie sobie ścisłej dyscypliny dąży do idealnej równowagi. Punktem podparcia dźwigni utrzymującej tę równowagę był taniec ludowy. Raz na tydzień uczestniczył w występach grupy tanecznej, co dawało mu odrobinę rozrywki i kontaktu z płcią odmienną. Jednakże pod koniec pierwszego roku na MIT zdarzyło się nieszczęście. Uszkodzenie kolana zmusiło go do porzucenia grupy tanecznej. Z początku wydawało mu się, że to nic wielkiego, że z czasem minie, i nawet więcej wolnego czasu niż poprzednio poświęcał tańczeniu, ograniczając nieco pracę w AI Lab. Jednakże z końcem lata, gdy ból kolana nie ustępował, a zajęcia miał się ponownie rozpocząć, zaczął się niepokoić. "Z kolaniem nie było ani ciut lepiej - wspomina, - co oznaczało, że muszę skończyć z tańczeniem. Byłem załamany".

Bez mieszkania i bez tańczenia jego życie towarzyskie skurczyło się, właściwie zanikło. Poczuł, że jego zdolność interakcji z niehakerami, a zwłaszcza z niehakerami płci żeńskiej uległa

znacznej atrofii. Po 16 tygodniach w AI Lab wyparowała z niego cała pewność siebie, którą zdążył zmagazynować przez cztery lata w Harvardzie.

"Czułem, że utraciłem całą energię - wspomina. Nie chciało mi się robić niczego, co nie było w danej chwili potrzebne. Energia potrzebna do zrobienia czegoś więcej zniknęła. Ogarnęła mnie czarna rozpacz".

Zatopiony bez reszty w pracy w AI Lab, wycofał się zupełnie ze świata. Przed październikiem 1975 porzucił MIT i nigdy tam nie powrócił. Hakerstwo, niegdyś jedynie hobby, stało się jego powołaniem.

Patrząc wstecz, uważa, że jego przeistoczenie się ze studenta studiów dziennych w pełnotętnego hakera było nieuniknione. Prędzej czy później głos przeznaczenia skłoniłby go do tej decyzji, tłumiąc zainteresowanie innymi zawodowymi poczynaniami. "Do fizyki i matematyki nigdy nie wniósłbym nic poważnego - wspomina swoje wysiłki z czasów przed uszkodzeniem kolana. - Mógłbym być nawet dumny, że coś tam zrobiłem w jednej lub drugiej dziedzinie, ale właściwie nie wiedziałem, do czego to ma prowadzić. Nie wiedziałem, od czego mam zacząć. W przypadku oprogramowania widziałem, co powinienem robić i jak działać, by napisać coś, co działa i jest pożyteczne. Przynosiło mi to satysfakcję popychającą do dalszego działania."

Stallman nie był pierwszym, dla którego hackerstwo było przyjemnością. Wielu hakerów z AI Lab chlubiło się tym samym i również przerwało studia. Większość miała jakieś stopnie naukowe z matematyki lub inżynierii elektrycznej, które z trudem zdobyła, podporządkowując się wymogom akademickiej kariery i zawodowych ambicji. Przyszli do AI Lab dla czystej przyjemności rozwiązywania problemów, którymi jeszcze nikt się nie zajmował. Na podobieństwo świętego Tomasza z Akwinu, którego długie i intensywne rozważania teologiczne prowadził do wizji nadprzyrodzonych, hakerzy osiągalni wewnętrzne stany transcendentne przez bardzo silne skupienie myśli i

wyczerpanie fizyczne. Choć Stallman, jak inni hakerzy, unikał narkotyków, lubił "haj", który nadchodził pod koniec 20-godzinnego programowania w transie.

Ale najwięcej radosnych emocji niosło uczucie samorealizacji. Hakerstwo było dla Stallmana rzeczą naturalną.

Przyzwyczajony od dziecka do długiego nocnego ślęczenia nad książkami, bez trudu znosił wielogodzinną pracę prawie bez snu. Będąc społecznym wyrzutkiem już w wieku 10 lat, nie miał kłopotów z pracą w samotności. Jako matematyk z wrodzoną zdolnością logicznego myślenia i przewidywania posiadał doskonałą umiejętność rozwiązywania problemów, w których grzęźły umysły innych hakerów.

"Był niezwykle - wspomina Gerald Sussman, wykładowca w MIT, niegdyś prowadzący badania w AI Lab. - Myślał jasno i jasno przedstawiał zamierzenia". To właśnie Sussman zatrudnił Stallmana na stanowisku asystenta naukowego w roku 1975.

Projekt był złożony i AI Lab miał o przygotować program analizujący diagramy obwodów. Wymagał nie tylko doskonałej znajomości poleceń Lispu, języka zbudowanego specjalnie dla potrzeb AI Lab, lecz również rozumienia i przewidywania, w jaki sposób różni ludzie mogą podchodzić do tego samego zagadnienia.

Jeżeli praca przy wykonywaniu oficjalnych projektów, w rodzaju programu Sussmana do automatycznego analizowania obwodów, pozostawiał anieco wolnego czasu, zajmował się programowaniem dla przyjemności. Hakerską ambicją było ulepszanie oprogramowania wykorzystywanego w AI Lab, a największym zrobionym dla przyjemności projektem Stallmana był używany w tym czasie w laboratorium program edycyjny TECO.

Historia jego pracy nad TECO w roku 1970 jest nierozzerwalnie związana z późniejszym przywództwem ruchu wolnego oprogramowania. Był to historyczny etap komputerowej ewolucji, więc trzeba go omówić, choćby w zarysie. W latach pięćdziesiątych i sześćdziesiątych, gdy komputery zaczęły

pojawiać się na uniwersytetach, programowanie było o działaniu niezwykle abstrakcyjnym. Aby skomunikować się z maszyną, programista przygotowywał plik kart dziurkowanych, z których każda reprezentowała jakieś polecenie. Potem przygotowany plik oddawało się administratorowi centralnego systemu, który według kolejności umieszczał go w czytniku maszyny. W odpowiedzi komputer wypływał nowy plik kart dziurkowanych, które programista musiał potem odcyfrować. Taki sposób działania, nazywany "procesem wsadowym", był niewygodny i czasochłonny. Był także denerwujący ze względów ambicjonalnych. Jedną z przyczyn wrodzonej hakerskiej awersji do centralizacji był a władza dzierzona przez administratorów systemu, którzy decydowali o kolejności wykonywania jobów¹⁰.

W roku 1962 naukowcy i hakerzy z MIT rozpoczęli realizację mającego zaradzić tej frustracji, projektu MAC11, z którego potem wyrósł o AI Lab. Dzielenie czasu, z początku nazywane "kradzieżą czasu", pozwoliło wielu programistom jednocześnie korzystać z możliwości operacyjnych komputera. Interfejsy dalekopisowe pozwoliły na komunikowanie się z komputerem przez wpisywanie zwykłego tekstu, zamiast karmienia go stosami dziurkowanych kart. Programista mógł teraz wpisywać kolejne wiersze poleceń i odczytywać odpowiedzi generowane przez maszynę.

W późnych latach sześćdziesiątych projektowanie interfejsów znacznie się rozwinęło. W słynnym wykładzie z roku 1968 Doug Engelbart, naukowiec ze Stanford Research Institute¹², pokazał prototyp nowoczesnego interfejsu graficznego. Łącząc kablem telewizor z komputerem i przyczepiając dodatkowe narzędzie wskazujące, które nazwał myszą, stworzył system nawet bardziej interaktywny od opracowanego w MIT systemu dzielenia czasu. Traktując ekran wyświetlający jako rodzaj szybkiej drukarki, system Engelbarta pozwalał użytkownikowi na przesuwanie kursora po ekranie i uaktualnianie jego pozycji przez komputer w czasie rzeczywistym.

Takie innowacje potrzebowały kolejnych dwóch dekad, by przeniknąć na rynek. W latach siedemdziesiątych rozpoczął się proces zastępowania dalekopisów przez terminale wyposażone w monitory, co dawało potencjalne możliwości korzystania wyświetlania obrazu na całym ekranie zamiast dotychczas stosowanej edycji linia po linii.

Jednym z pierwszych programów, który korzystał z możliwości pełnoekranowego wyświetlania informacji, był TECO stworzony przez AI Lab w MIT. TECO (skrót od Text Editor and Corrector) był dokonaną przez hakerów adaptacją programu dalekopisowego działającego w systemie linia po linii na PDP-6.

TECO stanowił istotny postęp w stosunku do edytorów starego typu, lecz nadal nie był wolny od wielu ich wad. Aby stworzyć i edytować dokument, programista musiał napisać szereg poleceń definiujących każdą zmianę edycyjną. Był to proces abstrakcyjny. W przeciwieństwie do współczesnych edytorów, uaktualniających tekst po każdym uderzeniu klawisza, aby coś zmienić, należało wpisać rozszerzony ciąg poleceń edytujących zakończonych dodatkowym ciągiem znaków oznaczającym "koniec polecenia". Z upływem czasu hakerzy nabrali wystarczającej wprawy, aby za pomocą tego narzędzia pisać całe dokumenty, ale Stallman przyznał później, że działanie to wymagało o specjalnych uzdolnień i sprawności przypominających umiejętność gry w szachy z zawiązanymi oczami¹³.

Aby ułatwić sobie pracę, hakerzy z AI Lab zbudowali system pozwalający na wyświetlanie tekstu w trybie "źródłowym" i "wynikowym" na dwóch częściach podzielonego ekranu. Innowacja była użyteczna, ale kłopot sprawiało przełączanie się między trybami.

W tym czasie TECO nie był jedynym pełnoekranowym monitorem. Podczas wizyty w Stanford Artificial Intelligence Lab¹⁴ w roku 1976 Stallman zobaczył edytor o nazwie E, którego cechą naturalną było uaktualnianie obrazu po każdym

naciśnięciu klawisza. Zgodnie z nazewnictwem obowiązującym w roku 1970, był to jeden z pierwszych rudymenarnych edytorów typu WYSIWYG. Jest to skrót od "what you see is what you get¹⁵" i oznacza, że użytkownik może manipulować plikiem, przesuwając tekst do tyłu i do przodu, zupełnie inaczej niż w przypadku edytora, w którym wynik można zobaczyć dopiero na końcu¹⁶.

Stallman był pod wrażeniem tego hakerskiego osiągnięcia i zastanawiał się, w jaki sposób po powrocie do MIT podobną funkcjonalność nadać TECO. Znalazł TECO wzbogacone o funkcję Ctrl-R napisaną przez Carla Mikkelsona i nazwaną tak od dwuklawiszowej kombinacji uruchamiającej, pozwalającej na przełączanie TECO z abstrakcyjnego trybu wykonywania poleceń w bardziej intuicyjny tryb reakcji na każde uderzenie klawisza. Stallman w subtelny ale znaczący sposób przekształcił tę funkcję. Umożliwił uruchamianie za pomocą kombinacji klawiszy innych poleceń tekstowych lub "makr". Trik Stallmana pozwalał na zapamiętanie raz napisanego polecenia i wielokrotne jego uruchamianie za pomocą kombinacji klawiszy. Rozwiązanie Mikkelsona podnosiło TECO na poziom WYSIWYG, a rozwiązanie Stallmana szło o jeden krok dalej, czyniąc z TECO programowalny edytor WYSIWYG. "To był przełom" - mówi Guy Steele, jeden z ówczesnych hakerów z AI Lab.

Stallman pamięta, że wprowadzenie makr wywołało eksplozję następnych innowacji. "Każdy pisał własny zbiór ulepszonych poleceń edytora, dodając polecenia wykonania wszelkich czynności, które wydawały mu się potrzebne". Wspomina, że "ludzie puszczali je w obieg, by każdy mógł coś dodać lub ulepszyć, dzięki czemu polecenia te stawały się coraz potężniejsze i miały coraz większy zakres działania. Zbiór predefiniowanych poleceń stawał się z wolna kolekcją programów tworzących cały system".

Wprowadzenie makr okazało się bardzo przydatną innowacją. Tak wiele osób zaczęło wbudowywać makra do swych wersji

TECO, że wkrótce w wyniku tej makromanii sam edytor stał się rzeczą mniejszej wagi. "Zaczęliśmy o nim myśleć raczej jak o języku programowania niż o edytorze" - mówi Stallman.

Doświadczeni użytkownicy dla własnej przyjemności dopracowywali szczegóły i zgłaszali nowe pomysły.

W dwa lata po tej eksplozji szybkość wprowadzania zmian zaczęła przynosić efekty uboczne. Skutkiem gwałtownego rozwoju było wiele zmian zwiększających wartość i funkcjonalność edytora, lecz także komplikujących go nadmiernie. "Mieliśmy do czynienia z efektem wieży Babel" - twierdzi Guy Steele.

Jak twierdzi, groziło to zniszczeniem zamysłu leżącego u podstaw przedsięwzięcia. Zaprojektowany przez hakerów ITS17 ułatwiał dzielenie się wiedzą i wzajemne poprawianie tworzonych programów. Można było usiąść przy czyimś terminalu, otworzyć plik, nad którym pracował, i bezpośrednio w programie robić zmiany i komentarze. "Czasami, aby pokazać komuś, jak należy coś zaprogramować lub debuggować, najwygodniej było zrobić to bezpośrednio na jego terminalu" - tłumaczy Steele.

Możliwość wprowadzania makr w drugim roku istnienia zaczęła przytłaczać edytor, czyniąc go nieprzydatnym. W swej gorliwości dodawania nowych cech pełnoekranowych hakerzy tak dalece przekształcali własne wersje TECO, że zasiadłszy do czyjegoś terminala trzeba było poświęcić około godziny na zorientowanie się, które makropolecenia do czego służyły. Sfrustrowany Steele podjął się rozwiązania tego problemu. Zebrał razem cztery różne zbiory makr i zaczął kreślić schemat dokumentujący znaczenie najczęściej używanych makropoleceń. Przystępując do implementacji udokumentowanych cech, zainteresował przedsięwzięciem Stallmana.

"Najpierw zaglądał mi przez ramię, pytając, co robię" - wspomina Steele.

Dla Steele'a, spokojnego hakera, który rzadko bezpośrednio

rozmawiał ze Stallmanem, wspomnienie tej rozmowy nadal jest żywe. Podglądanie przez ramię, co kto robi, był ową AI Lab przyjętym zwyczajem. Stallman, główny opiekun TECO, uznał pracę Steele' a za interesującą i szybko przystąpił do jej skompletowania.

"Jak już wielokrotnie mówiłem, wykonałem 0,001 procenta implementacji, a resztę wykonał Stallman" - wspomina Steele ze śmiechem.

Nowa nazwa projektu, Emacs, była kurtuazyjnym gestem Stallmana, skrótem od ang. editing macros¹⁸, mającym przypominać ewolucyjny, transcendentny proces tworzenia makr w ciągu dwóch lat od eksplozji zainteresowania tym narzędziem. Było to również zapełnienie słownikowej luki, gdyż dotychczas nazwa żadnego programu pracującego w systemie ITS nie rozpoczynała się na literę "E". Wybierając Emacs, Stallman sądził, że pozwoli to na odwoływanie się do programu przez podawanie jedynie pierwszej litery nazwy. Po raz kolejny dał a osobie znać hakerska żądza maksymalnej efektywności. Rozwijając system standardowych makropoleceń, Stallman i Steele musieli wykonać polityczną akrobację na linie. Tworząc standardowy program, Stallman gwałcił fundamentalną hakerską zasadę "popieraj decentralizację". Musiał także wziąć w karby nieograniczoną dostępność czynienia wszelkich zmian, co było główną siłą napędową rozwoju TECO.

"Z jednej strony staraliśmy się zbudować jednolity system poleceń, a z drugiej - nie zamykać drogi do wprowadzania innowacji, ponieważ programowalność była rzeczą bardzo ważną" - wspomina Stallman.

Aby rozwiązać problem, Stallman, Steele i ich koledzy hakerzy David Moon i Dan Weinreib ograniczyli standaryzację jedynie do poleceń WYSIWYG, czyli sterujących sposobem wyświetlania tekstu na ekranie. Resztę wysiłku poświęcono na zachowanie tinkertoyowego¹⁹ stylu Emacs'a, pozwalającego na rozszerzanie funkcjonalności programu.

Przed Stallmanem stanął nowy trudny do rozwiązania problem:

jeżeli użytkownicy będą dokonywać zmian, nie informując o tym innych, znów pojawi się efekt wieży Babel. Wracając do hakerskiej idei dzielenia się wszelkimi informacjami, umieścił w kodzie źródłowym oświadczenie określające zasady używania Emacs'a. Każdy użytkownik mógł w programie dokonywać dowolnych zmian pod warunkiem udostępnienia reszcie społeczności wszystkich dokonanych rozszerzeń. Nazwał to "wspólnotą Emacs20". Tak jak TECO było czymś więcej niż edytorem, tak Emacs stał się czymś więcej niż programem. Dla Stallmana była to umowa społeczna. We wspomnieniu dokumentującym wczesną fazę projektu wyliczył zasady tego kontraktu. "EMACS był rozpowszechniany na podstawie zobowiązania do społecznego dzielenia się informacją, co oznaczało, że należało przesyłać informację o każdym wprowadzonym ulepszeniu, aby można je było włączyć do programu ku powszechnemu pożytkowi"²¹.

Nie wszyscy akceptowali te warunki kontraktu. Gwałtowny rozwój trwał przez następną dekadę, co doprowadziło do powstania wielu emacsopodobnych programów o różnym stopniu zgodności z oryginałem. W kilku pochodzenie od stallmanowskiego oryginału zaznaczono w dowcipny sposób za pomocą nazw rekursywnych, np. Sine (Sine is not Emacs)²², Eine (Eine is not Emacs) i Zwei (Zwei was Eine initially)²³. Jako przysięgły wyznawca hakerskiej etyki Stallman nie widział żadnego powodu, by za pomocą środków prawnych powstrzymać to twórcze zamieszanie. Fakt, że niektórzy potrafili użyć czyjegoś programu, nieco go ulepszyć i udostępniać pod inną nazwą, uważał za zdumiewające nieprzestrzeganie dobrych obyczajów.

Kolejny przykry incydent zakłócił rozwój hakerskiej społeczności. W roku 1979 Brian Reid zdecydował się na umieszczenie "bomby z opóźnionym zapłonem" w kodzie programu Scribe, umożliwiając firmie Unilogic ograniczanie dostępu do oprogramowania tym użytkownikom, którzy nie uiścili stosownej opłaty. Dla Stallmana była to złowieszcza

zapowiedź przyszłych zdarzeń. "Uznał to za postępowanie w stylu nazistowskim, najgorsze, z jakim się spotkał" - wspomina Reid. Późniejsze zdobycie sławy współtwórcy internetowego Usenetu²⁴ nie zmyło z Reida, przynajmniej w oczach Stallmana, hańby tamtej decyzji. "On twierdzi, że całe oprogramowanie powinno być wolne, a pomysł pobierania za nie opłaty jest zbrodnią przeciw ludzkości"²⁵.

Choć Stallman był bezsilny wobec prowadzonej przez Reida sprzedaży, udało mu się jednak ukrócić inne poczynania sprzeczne z hakerskim etosem. Jako główny depozytariusz kodu źródłowego wspólnoty Emacs zaczął politycznie wykorzystywać płynące stąd możliwości. Na ostatnich etapach konfliktu z administracją Laboratory for Computer Science w sprawie zabezpieczenia dostępu do systemu przez wprowadzenie haseł rozpoczął strajk polegający na odmowie przesyłania tym osobom najnowszych wersji Emacsa do czasu, aż zrezygnują z wprowadzenia zabezpieczeń. Ruch ten tylko trochę przyczynił się do utrwalenia jego reputacji ekstremisty, ale dał istotny efekt pośredni: wzniecił w społeczności dyskusję o podstawowych wartościach hakerskiej etyki.

"Wiele osób wściekało się na mnie, twierdząc, że jestem do nich wrogo nastawiony lub że ich szantażuję, co w pewnym sensie był prawdą - Stallman opowiadał później Stevenowi Levy'emu. - Działam przeciw nim siłą, ponieważ uważałem, że to, co robia, jest gwałtem zadawanym każdemu z nas²⁶".

Z upływem czasu Emacs stał się probierzem hakerskiej etyki. Zapewniona przez Stallmana elastyczność tego oprogramowania zachęcała do współpracy, nawet jej żądała. Użytkownicy, którzy nie nadążali za rozwojem Emacsa lub nie zgłaszali Stallmanowi wprowadzonych przez siebie poprawek, ryzykowali pominięcie przy informowaniu o kolejnym przełomie. A informacji przełomowych było dużo. Dwadzieścia lat później użytkownicy zmodyfikowali Emacsa na tak wiele sposobów, że bywa używany jako arkusz kalkulacyjny, kalkulator, baza danych i przeglądarka internetowa. To tylko fragment istnego

potopu różnych zastosowań, świadczący o wielkiej wszechstronności tego oprogramowania. "To właśnie było naszym celem - twierdzi Stallman. - Wszystko, co zostało w tym zawarte, jest jednocześnie cudowne i straszne".

Ocena innych ówczesnych współpracowników Stallmana z AI Lab jest znacznie łaskawsza. Hal Abelson, który jako student przed dyplomem współpracował ze Stallmanem w latach siedemdziesiątych, a potem wspierał go jako członek zarządu Fundacji Wolnego Oprogramowania, opisuje Emacs jako "twórbyłyskotliwy i doskonały". Twierdzi, że Stallman, dając możliwość dodawania własnych bibliotek i narzędzi bez wprowadzania bałaganu w ogólnej konstrukcji systemu, wytyczył drogę dalszego rozwoju projektów programistycznych opartych na prowadzonej na wielką skalę współpracy. "Jego struktura była wystarczająco krzepka, by wytrzymać ciężar luźnej współpracy ludzi z całego świata, wnoszących własne pomysły - powiada Abelson. - Nie sędzę, aby ktoś wcześniej dokonał czegoś podobnego"²⁷.

Guy Steele w podobny sposób wyraża uznanie. Obecny pracownik naukowy Sun Microsystems, pamięta Stallmana przede wszystkim jako "byłyskotliwego programistę zdolnego do produkowanie olbrzymiej objętości bezbłędnych kodów". Choć ich osobowości niezbyt do siebie pasowały, współpracowali wystarczająco długo, by Steele wyrobił sobie zdanie o stallmanowskim intensywnym stylu programowania.

Wspomina znaczący epizod z końca lat siedemdziesiątych, gdy postanowili wspólnie napisać funkcję "pięknego drukowania". Ta wymyślona przez Steele'a funkcja, uruchamiana za pomocą skrótu klawiszowego, powodowała przeformatowanie kodu źródłowego Emacsa, tak że stawał się czytelniejszy i zajmował mniej miejsca, co było rozszerzeniem możliwości WYSIWYG edytora. Była to funkcja o znaczeniu na tyle ważnym, by przyciągnąć uwagę Stallmana i wkrótce po napisaniu przez Steele'a wersji pierwotnej obaj postanowili zasiąść razem, by napisać wersję poprawioną.

"Zaczęliśmy pracę pewnego ranka - wspomina Steele. - Ja przy klawiaturze, a on obok mnie. Ustąpił mi miejsca przy klawiaturze, ale koniecznie chciał dyktować, co mam pisać." Sesja trwała 10 godzin. Przez cały czas, jak twierdzi Steele, żaden z nich nie oderwał się od pracy ani na moment. Pod koniec "hak" był gotowy i nie przekraczał 100 linii kodu. "Przez cały czas moje palce były na klawiaturze - wspomina Steele, - ale wydawało się, że pomysły nas obu sprawnie spływały na ekran. On mówił, co mam pisać, a ja pisałem". To, jak długo trwała ta sesja, Steele uświadomił sobie dopiero, gdy w końcu opuścił AILab. Stając przed budynkiem na Tech Square 545 ze zdumieniem zobaczył, że otaczają go nocne ciemności. Był przyzwyczajony do programistycznych maratonów, ale w tej sesji było coś niezwykłego. Praca ze Stallmanem zmusiła go do odcięcia się od wszystkich bodźców zewnętrznych i skupienia całej mentalnej energii na wykonywanym zadaniu. Patrząc wstecz, uważa, że Stallman stapiał w jedno myśli swoje i partnera, napawając go jednocześnie radością i przerażeniem. "Gdy to się skończyło, pomyślałem, że choć było to wspaniałe doświadczenie, nie chciałbym go przeżywać ponownie."

Przypisy

1 Patrz: Josh McHugh "For the Love of Hacking" Forbes (10 sierpnia 1998):

<http://www.forbes.com/forbes/1998/0810/6203094a.html>.

2 Stallman w r. 1974 otrzymał w Harvardzie tytuł Bachelor of Arts (BA) - patrz:

<http://www.stallman.org/home.html#serious>, co mniej więcej odpowiada polskiemu licencjatowi. Internetowa Encyklopedia PWN (http://encyklopedia.pwn.pl/5310_1.html) opisuje ten stopień następująco:

BAKALAUREAT [łac.]: najniższy stopień nauk. nadawany od XIII w. przez uniw. eur., a obecnie gł. przez uniw. anglosaskie

(bachelor) po 3 latach studiów; np. Bachelor of Arts (BA), Bachelor of Science (BSc).

Autor podaje, że na MIT Stallman wstąpił jako "grad student", co może oznaczać studia kończące się dyplomem magisterskim (Master of Science - MSc, Master of Arts - MA) lub doktorskim (Doctor of Philosophy - PhD). Ponieważ w Harvardzie nie otrzymał magisterium, należy wnioskować, że ten stopień miał zdobyć na MIT, więc były to studia odpowiadające ostatnim latom naszych studiów magisterskich. W uproszczeniu

tl umacze to jako studia magisterskie - przyp. tl um

3 Patrz: Stallman (1986).

4 W późnych latach sześćdziesiątych Joseph Weizenbaum, zatrudniony

w MIT, stworzył program komputerowy ELIZA, który z grubsza imitował psychoterapeutę przeprowadzającego wstępną rozmowę z pacjentem. Wielu ludzi z MIT, związanych z informatyką, angażował o się emocjonalnie w rozmowy z komputerem, dzieląc się swymi najbardziej intymnymi myślami. Weizenbaum, zaniepokojony wzrastającą wśród naukowców, a nawet wśród "zwykłych" ludzi tendencją do traktowania istoty ludzkiej jak maszyny, napisał książkę broniącą poglądu, iż ludzie są czymś więcej niż tylko "przetwarzaczami informacji". Computer Power and Human Reason (w swobodnym tl umaczeniu "potęga komputera i rozum człowieka" lub "potęga komputera i czynnik ludzki") jest dzisiaj uważana za pozycję klasyczną - przyp. tl um.

5 "Bum" według słowników to wł óczęga; wagabunda; koczownik; tramp ale także ktoś bez reszty poświęcony jakiejś działalności. Z kontekstu wynika, że Weizenbaum miał na myśli połączenie tych cech, ale nie znam polskiego słowa o takim znaczeniu - przyp. tl um.

6 Patrz: Joseph Weizenbaum, Computer Power and Human Reason: From Judgment to Calculation (W.H. Freeman, 1976): s. 116.

7 William Bradford Shockley (1910-1989), fizyk

amerykański. Autor prac dotyczących własności magnetycznych i półprzewodnikowych ciał stałych. W 1949 wraz z J. Bardeenem i W.H. Brattainem wynalazł tranzystor, za co wszyscy trzej w roku 1956 otrzymali Nagrodę Nobla - przyp. tłum.

8 Richard Philips Feynman (1918-1988), wybitny amerykański fizyk-teoretyk, członek amerykańskiej Akademii Nauk. Jeden z autorów pełnego sformułowania elektrodynamiki kwantowej, za co wraz z J. Schwingerem i S. Tomonagą otrzymał w roku 1965 Nagrodę Nobla - przyp. tłum.

9 Murray Gell-Mann (ur. 1929), amerykański fizyk-teoretyk pracujący Kalifornijskim Instytucie Technologicznym, zajmujący się fizyką cząstek elementarnych. Wprowadził pojęcie kwarku. Po potwierdzeniu jego modelu dzięki odkryciu postulowanej przez niego cząstki: barionu Λ^0 - otrzymał w roku 1969 Nagrodę Nobla - przyp. tłum.

10 Oczywiście, poprawnie po polsku należałoby napisać "plików wsadowych" (a był yto rzeczywiste pliki kart, a nie obecne pseudopliki, będące ciągiem znaków zapisanych gdzieś tam na dysku), ale w rzeczywistości nikt tak nie mówił. Z własnego doświadczenia pamiętam, że mówiło się o "wrzucaniu jobów do czytnika". Job to było coś więcej niż dzisiejszy plik danych, zawierał bowiem polecenia wstępne, wśród nich identyfikator programisty, czasem numer jego konta, potem polecenia uruchamiające kompilator bądź inny program mający obrabiać dane zapisane we właściwym pliku, następnie znak początku pliku danych, plik danych i znak ich końca. Oczywiście, były yjoby bardziej skomplikowane, np. zawierające kilka plików danych - przyp. tłum.

11 Patrz przypis 12. w rozdziale 4. - przyp. tłum.

12 W skrócie SRI - niezależny pozarządowy instytut badawczy ufundowany przez przemysł ołców z Zachodniego Wybrzeża i Stanford University, początkowo jako dependencja uniwersytetu. W roku 1970 uniezależnił się, a w roku 1977 zmienił nazwę na Stanford Research Institute International. W

SRI prowadzone są głównie badania dotyczące najnowszych technologii. Pierwszą próbę transmisji danych przez sieć ARPANET między SRI i UCLA (University of California, Los Angeles) 29 października 1969 uznaje się za narodziny Internetu - przyp. tłum.

13 Patrz: Richard Stallman EMACS: The Extensible, Customizable, Display Editor, AI Lab Memo (1979). Uaktualniona wersja w formacie html, z której cytuję, jest dostępna pod adresem:

<http://www.gnu.org/software/emacs/emacs-paper.html>

14 Stanfordzkie laboratorium sztucznej inteligencji - przyp. tłum.

15 Otrzymasz to, co widzisz - przyp. tłum.

16 Patrz: Richard Stallman Emacs the Full Screen Editor (1987): <http://www.lysator.liu.se/history/garb/txt/87-1-emacs.txt>. Dalsze informacje o rozbudowie TECO pochodzą z tego samego źródła.

17 The Incompatible Timesharing System (ITS). Por. przypis 17. w rozdziale 4. i tekst, do którego on się odnosi - przyp. tłum.

18 Makra edycyjne lub makra edytujące - przyp. tłum

19 Tinker Toy (Tinker-Toy, Tinkertoy lub tinkertoy), czyli "zabawka druciarza" - znana od roku 1913 zabawka-układanka składająca się z patyczków i elementów łączących, pozwalająca na rozwijanie wyobraźni dziecka. Jest to pojęcie przysłowiowe, określające układankę dającą pole wyobraźni - znalazłem w Internecie reklamę firmy, która drewniane domy, sprzedawane w częściach do samodzielnego montażu, reklamowała jako Tinker Toy dla dorosłych. Klasyczny Tinker Toy wygląda tak jak na rysunku poniżej - przyp. tłum.

20 Emacs Commune, co można przełożyć jako wspólnota, społeczność lub komuna Emacs albo emacsowa. W Polsce temat nie jest często poruszany, więc nie ma ogólnie przyjętego tłumaczenia - przyp. tłum.

21 Patrz Stallman (1979): #SEC34.

22 Sine nie jest Emacs - przyp. tłum.

23 Zwei było na początku Eine - przyp. tłum.

24 Usenet - ogólnosiwiatowy komputerowy system dyskusyjny (inaczej: rozproszony biuletyn elektroniczny) wykorzystujący do transmisji danych Internet i inne sieci. Dyskusje odbywają się w grupach tematycznych
- przyp. tłum.

25 W roku 1996 w wywiadzie dla internetowego magazynu MEME Stallman podał sprzedaż Scribe' a jako rzecz przykłą, ale powstrzymał się od wymienienia nazwiska Reida. "Problemem jest to, że nikt nie ocenił i nie ukarał tego studenta za to, co zrobił" - powiedział . - "W rezultacie inni nabrali ochoty, by iść w jego ślady". Patrz: MEME 2.04

<http://memex.org/meme2-04.html>

26 Patrz: Steven Levy Hakerzy (Penguin USA, 1984), s. 419.

27 Pisząc ten rozdział ,skupiałem się bardziej na zagadnieniach społecznych niż programistycznych. Więcej informacji o tych drugich można znaleźć we wspomnieniu Stallmana z roku 1979. Polecam szczególnie rozdział "Research Through Dvelopment of Insatlld Tools" (#SEC27). Jest to tekst zrozumiał y również dla osób o wykształ ceniu nietechnicznym i rzuca światł ona ściśł epowiązanie politycznej filozofii Stallmana z jego filozofią projektowania oprogramowania. Oto przykł ad:

EMACS nie mógł powstać jako coś z góry starannie zaprojektowanego. Takie postępowanie jest możliwe jedynie wówczas, gdy od początku jest znany cel działania i od początku są znane potrzeby, które należy zaspokoić. Ani ja, ani nikt inny nie wyobrażaliśmy sobie tak rozbudowanego edytora, dopóki nie został on stworzony. Przed poznaniem jego możliwości nie docenialiśmy także jego wartości. EMACS istnieje, ponieważ ośmielił em się wykonać kilka pożytecznych ułatwień na drodze, której koniec znajdował się daleko poza zasięgiem mojego wzroku.

7. TRUDNY WYBÓR MORALNY

27 września 1983 roku programiści logujący się do usenetowej grupy net.unix-wizards natykali się na niezwykle komunikat. Wysłany tuż po północy, a ściślej o 00:30, sygnowany przez rms@mit-oz, w wierszu tematu zawierał zwięzły ale przyciągający uwagę komunikat: "New UNIX implementation" (nowa implementacja Uniksa). Jednakże nie było to wprowadzenie do nowej wersji Uniksa - otwierający wiadomość akapit zawierał komunikat:

Poczynając od Thanksgiving², zamierzam rozpocząć pisanie pełnego, zgodnego z Uniksem systemu operacyjnego GNU (od Gnu' sNot Unix), by udostępnić go wszystkim chętnym za darmo. Pilnie potrzebne jest wsparcie w postaci poświęconego czasu, pieniędzy, oprogramowania i sprzętu³.

Dla doświadczonych twórców Uniksa brzmiało to jak mieszanina idealizmu i arogancji. Autor nie tylko przyrzekał zbudowanie od nowa w pełni funkcjonalnego systemu uniksowego, lecz również ulepszenie go w wielu miejscach. Zgodnie z zapowiedzią autora, nowy system GNU miał zawierać wszystkie zwyczajowe elementy - edytor tekstowy, program powłoki umożliwiający uruchamianie zgodnych z Uniksem aplikacji, kompilator i "parę innych rzeczy". Dodatkowo miał być wyposażony w kilka pożądanых elementów, których inne systemy uniksowe jeszcze wówczas nie oferowały: graficzny interfejs użytkownika, zaprogramowany w języku Lisp, odporny na załamanie system plików i protokoły sieciowy zbudowany zgodnie z wewnętrznym systemem sieciowym MIT.

"GNU będzie pozwalał na uruchamianie programów uniksowych, ale nie będzie identyczny z Uniksem - pisał autor. - Na podstawie naszych doświadczeń z innymi systemami operacyjnymi dokonamy wszystkich przydatnych poprawek." Przewidując sceptycyzm części adresatów komunikatu, autor obok opisu systemu operacyjnego umieścił własną notę biograficzną zatytułowaną "Kim jestem?"

Nazywam się Richard Stallman i jestem pomysłodawcą oryginalnego i często imitowanego edytora EMACS. Obecnie pracuję w Laboratorium Sztucznej Inteligencji w MIT. Brałem aktywny udział w pracach nad kompilatorami, edytorami, debuggerami, interpretatorami poleceń, a także systemem ITS (ang. Incompatible Timesharing System⁴) oraz systemem operacyjnym maszyn LISP⁵. Byłem jednym z pionierów tworzenia, niezależnych od terminali programów obsługi wyświetlania informacji w systemie ITS, a także wykonałem implementacje odpornego na załamania systemu plików oraz dwóch systemów okienkowych na maszynach lispowych. Jak było do przewidzenia, podana przez Stallmana nazbyt optymistyczna data rozpoczęcia projektu GNU nie została dotrzymana. Jednakże do stycznia 1984 roku Stallman rzeczywiście zaczął wypełniać daną obietnicę i stał się osobą znaczącą w środowisku twórców oprogramowania dla systemów uniksowych. Dla architekta oprogramowania, który wyrósł na systemie ITS, było to jak projektowanie sklepienia przez projektanta pałaców mauretańskich. Pomimo to budowanie systemu operacyjnego podobnego do Uniksa przynosiło ukryte korzyści. ITS był systemem potężnym, lecz miał swoją piętę achillesową: hakerzy z MIT zaprojektowali go, wykorzystując wiele cech charakterystycznych dla budowanych przez firmę DEC⁶ komputerów PDP. Gdy na początku lat osiemdziesiątych zarządzający AI LAB zdecydowali o usunięciu potężnego komputera PDP-10, stworzony przez hakerów system operacyjny w jednej chwili stał reliktem minionej epoki. Unix przeciwnie, został zaprojektowany z myślą o znacznej mobilności, miał przed sobą perspektywę o wiele dłuższego życia. W wersji pierwotnej stworzony przez młodych naukowców z AT&T⁷, wymknął się spod firmowej kontroli i zadomowił na dobre w zawsze biednym świecie komputerowych sieci akademickich. Dysponujący mniejszymi środkami niż ich koledzy z MIT, twórcy Uniksa przystosowali system do używania na przeróżnym sprzęcie: od 16-bitowych PDP-11

używanych w AI Lab do wykonywania pomniejszych zadań, po duże 32-bitowe komputery typu mainframe⁸, jak VAX 11/780. Przed rokiem 1983 kilka firm, łącznie z zasłużoną Sun Microsystems, przymierzało się do stworzenia nowej generacji mikrokomputerów zwanych stacjami roboczymi, które miały wykorzystywać możliwości nowego coraz bardziej powszechnego systemu.

Aby ułatwić sobie zadanie, projektanci zasadniczej części Uniksa, nadający ogólny kierunek pracom, stworzyli między maszyną i systemem dodatkową abstrakcyjną warstwę.

Hakerzy z AI Lab, tworząc system ITS maksymalnie wykorzystali możliwości sprzętu, zaś twórcy Uniksa zrezygnowali z przykrawania systemu operacyjnego do zasobów sprzętowych i wykorzystywania płynących stąd pożytków.

Wszechstronność systemu przełożyli nad doraźne korzyści.

Skupiając się bardziej na wprowadzeniu standardów i specyfikacji łączących różne elementy systemu niż na tworzeniu samych komponentów, stworzyli system, który mógł być szybko zmodyfikowany i dostosowany do "gustu" dowolnej maszyny. Jeżeli użytkownik miał kłopoty z jakąś częścią systemu, można ją było usunąć, naprawić albo wymienić na inną. Choć z tego powodu cierpiał estetyka systemu, zwiększał o to jego elastyczność i opłacalność, gdyż łatwo było o przystosować go do nowych wymagań⁹.

Koniec ITS-a, wychuchanego i tak długo hołubionego przez hakerów z AI Lab, przyczynił się do podjęcia przez Stallmana decyzji o budowie nowego systemu GNU. Głęboko przeżył rozstanie z ITS-em. Po epizodzie z drukarką laserową Xeroksa był to kolejny dowód na to, że hakerska kultura AI Lab traci swą odporność na zewnętrzne działanie świata biznesu.

Wraz z kodem tworzącym ITS zamarł ykorzenie, z których ten system wyrósł. W latach po zakończeniu wojny wietnamskiej znacznie zmniejszyły się wydatki na obronę, będące źródłem finansowania głównej części komputerowych prac badawczych. Desperacko poszukując nowych funduszy, laboratoria i

uniwersytety zwrócił się do sektora prywatnego. W przypadku AI Lab pozyskanie prywatnych inwestorów nie było trudne. Gniazdo, w którym wykuło się wiele ambitnych powojennych projektów komputerowych, szybko stało się wylegarnią nowych technologii. Zanim nastał rok 1980, wielu hakerów z AI Lab zaczęło dzielić czas między prace badawcze w instytucie i projekty komercyjne.

To, co zdawał o się dawać równe korzyści obu stronom, umożliwiając hakerom udział w najlepszych projektach i dostęp do najnowszych technologii, wkrótce okazało się diabelskim pomysłem. Im więcej czasu hakerzy poświęcali na awangardowe komercyjne projekty, tym mniej mogli go przeznaczać na utrzymanie wyrafinowanego oprogramowania samego AI Lab. Wkrótce firmy zaczęły bezpośrednio angażować hakerów, aby w pełni dysponować ich czasem i bez reszty skupić ich uwagę na interesujących je zagadnieniach. Z powodu zmniejszającej się liczby hakerów wszelkie zmiany i naprawy oprogramowania w laboratorium trwały coraz dłużej. Co gorsze, Stallman stwierdził, że nastąpiła zmiana pokoleniowa. Hakerzy, którzy dotychczas stanowili w AI Lab decydującą mniejszość, zaczęli się wykruszać, a liczba "profesorów i studentów nie pałających zbyt wielką miłością do PDP-10 pozostawała niezmieniona"¹⁰.

Punktem przełomowym był rok 1982. Wtedy to administracja uczelni zdecydowała o wymianie PDP-10 na komputer nowej generacji. Firma Digital, dostawca PDP-10, zaprzestał produkcję komputerów z tej serii, ale zaoferował apoteńny komputer mainframe o nazwie KL-10. Jednakże nowa maszyna wymagała poważnych przeróbek systemu ITS, gdyby hakerzy zdecydowali się na dalsze korzystanie z niego. Rzeczą straszną byłaby utrata przez AI Lab intelektualnej masy krytycznej. Ubyło talentów programistycznych, a członkowie rady wydziału naciskali na zakup Twenexu, komercyjnego systemu operacyjnego stworzonego przez Digital. Nieliczni pozostali hakerzy musieli się dostosować.

Rada wydziału orzekła, że "brak hakerów mogących utrzymać własny system może doprowadzić do nieszczęścia, więc konieczne należy zakupić system komercyjny". Jak po latach wspomina Stallman, uznano, że "firma zapewni jego działanie". Potem okazało się, że nie mieli racji, ale był o już za późno. Z początku hakerzy uznali Twenex za kolejny symbol autorytaryzmu, który aż się prosi o obalenie. Samo nazwanie systemu Twenexem było protestem przeciw niemu. Przez DEC nazywany oficjalnie TOPS-20, był następcą TOPS-10, komercyjnego systemu opracowanego dla komputerów PDP-10. Najpierw w firmie Bolt, Beranek & Newman¹¹ powstała nowa, ulepszona wersja nazwana Tenex i na jej podstawie powstał TOPS-20¹². "Temu systemowi było daleko do szczytów¹³, więc żadną miarą nie zamierzałem go tak nazywać, więc zdecydowałem wstawić "w" do nazwy Tenex i nazwać system Twenexem" - wspomina Stallman, który wymyślił nową nazwę. Maszyna, na której działał Twenex/TOPS-20, miała własne szydercze przezwisko: Oz, które wzięło się stąd, że jej terminale nie działały samodzielnie i każdy musiał być wsparty przez małe PDP-11. Jak podaje jedna z hakerskich legend, pewien haker, zobaczywszy po raz pierwszy to sprzężenie KL-10 z PDP-11, skojarzył informację wstępną wyświetlaną na ekranie z napuszczonym oświadczeniem Czarnoksiężnika z Krainy Oz i zaczął nucić: "Jam jest wielki i potężny Oz. Nie zwracaj uwagi na małe PDP-11 ukryte za konsolą"¹⁴. Jeżeli hakerów rozbawiło pierwsze zetknięcie z KL-10, to miny im zrzedły po ujrzeniu systemu Twenex. Nie tylko wbudowano w system zabezpieczeń, także wszystkie narzędzia i aplikacje zaprojektowano z myślą o zabezpieczeniach. Poprzednia zabawa w kotka i myszkę z hasłami wprowadzanymi przez Laboratorium Bezpieczeństwa Komputerowego teraz przekształciła się w totalną wojnę z systemem zarządzania. Administratorzy systemu argumentowali, że bez zabezpieczeń system Oz byłby narażony na częstsze załamania. Hakerzy twierdzili, że przed załamaniami systemu najlepiej można się

zabezpieczyć przez gruntowne przejrzenie i poprawienie kodu. Niestety, liczba hakerów chętnych i zdolnych do wykonania tego zadania zmalała tak znacznie, że zwycięstwo administratorów stało się oczywiste.

Zdobywając różnymi sposobami hasła i irozmyślnie doprowadzając do załamania systemu, by wśród szczątków poszukiwać dowodów, Stallman z powodzeniem udaremniał zapędy administratorów do przejęcia kontroli nad sprzętem. Po jednym udaremnionym coup d'etat wystosował ostrzeżenie do wszystkich pracowników AI Lab:

"Została podjęta kolejna próba przejęcia władzy - pisał . - Na razie siły magnatów zostały pokonane". Aby ukryć swoją tożsamość, ostrzeżenie podpisał "Radio wolny OZ".

Niewielu to zmyliło. Do roku 1982 awersja Stallmana do haseł i tajności stała się powszechnie znana również poza AI Lab i ludzie z zewnątrz używali jego konta jako punktu wejścia do ARPANET-u, badawczej sieci komputerowej, która stała się fundamentem przyszłego Internetu. Jednym z takich "turystów" we wczesnych latach osiemdziesiątych był Don Hopkins, programista z Kalifornii. Uczył się on, czerpiąc obficie ze skarbnicy wiedzy, jaką był wychwalany system ITS, do którego można było wejść z zewnątrz, wpisując jako login inicjały RMS, a następnie te same trzy litery jako hasło.

"Jestem dożywotnio wdzięczny MIT za to, że mogłem wraz z wieloma innymi korzystać za darmo z ich komputerów - mówi Hopkins. - Dla wielu ludzi miało to wielkie znaczenie."

Była to tak zwana polityka "turystyczna" otwarcie tolerowana przez władze MIT w czasach systemu ITS15. Skończyła się, gdy Oz stał się głównym wejściem laboratorium do sieci ARPANET. Na początku Stallman nadal używał swego loginu również jako hasła, aby użytkownicy zewnętrzni mogli łatwo robić to samo. Jednakże z upływem czasu kruchość systemu Oz skłoniła administratorów systemu do zataśmowania dostępu dla użytkowników z zewnątrz. Obawiano się, że przez czysty przypadek lub przez czyjąś złośliwość może dojść do pełnego

załamania systemu. Gdy zażądano, by Stallman przestał podawać swoje hasło do publicznej wiadomości, on, powołując się na wyznawane przez siebie zasady etyczne, odmówił i zrezygnował z korzystania z komputera Oz.

"Gdy hasła po raz pierwszy pojawiły się w AI Lab, nie zmieniłem zdania i nadal nie widziałem potrzeby ich stosowania - twierdził Stallman. - Ponieważ nie wierzę w konieczność chronienia zasobów komputera, nie mogłem wspierać wprowadzenia reżimu bezpieczeństwa".

Jego odmowa dostosowania się do wymagań Oz był a odzwierciedleniem, narastającego w ciągu lat osiemdziesiątych, napięcia między grupą hakerów i administracją AI Lab. Było to jednak niczym w porównaniu z konfliktem, który wybuchł wewnątrz samej społeczności hakerskiej. Jeszcze przed przybyciem KL-10 podzieliła się ona na dwa obozy. Pierwszy skupił się wokół firmy programistycznej Symbolics, Inc., drugi - wokół rywalizującej z nią Lisp Machines, Inc. (LMI). Obie firmy dążyły do wprowadzenia na rynek maszyny lispowej, narzędzia mającego służyć do pełnego wykorzystania korzyści płynących ze stosowania języka programowania Lisp.

Stworzony przez Johna McCarthy'ego, pioniera badań nad sztuczną inteligencją, pracownika naukowego MIT w latach pięćdziesiątych, Lisp jest językiem programowania świetnie przystosowanym do wykonywania zadań wymagających skomplikowanego sortowania. Nazwa jest skrótem od List Processing¹⁶. Po odejściu McCarthy'ego do Stanford Artificial Intelligence Laboratory hakerzy z MIT dopracowali język, przekształcając go w lokalny dialekt MACLISP. "MAC" pochodziło od finansowanego przez DARPA¹⁷ projektu badawczego MAC¹⁸, z którego wyrosło AI Lab i Laboratory for Computer Science. Programiści z AI Lab pod kierownictwem arcyhakerka Richarda Greenblatta, używając jedynie języka Lisp, zbudowali w latach siedemdziesiątych pełny system operacyjny, który nazwali Lisp Machine Operating System¹⁹. Do roku 1980 z projektu Lisp Machine powstały dwie firmy

komercyjne: Symbolics kierowana przez Russela Noftskera, niegdyś administratora AI Lab, oraz Lisp Machines, Inc. z Greenblattem na czele.

Oprogramowanie Lisp Machine zostało stworzone na zasadach hakerskich, co oznacza, że choć jego właścicielem pozostawało MIT, zgodnie z hakerskim obyczajem, mogło być swobodnie kopiowane. Taki system ograniczał korzyści rynkowe, które firmy miały nadzieję osiągnąć, uzyskując licencję od MIT i usiłując wprowadzić to oprogramowanie na rynek jako unikalne. Aby zarabiać, firmy dążyły do uzupełnienia systemu Lisp Machine dodatkowymi elementami, które mogły okazać się atrakcyjne dla użytkowników. Do tworzenia tych dodatków, powstających poza MIT, zatrudniały hakerów z AI Lab.

Sz szczególnie agresywnie działała firma Symbolics. Do końca roku 1980 już 14 hakerów z AI Lab było tam zatrudnionych w roli konsultantów na części etatu, pracując nad rozwojem własnej wersji Lisp Machine. Reszta, za wyjątkiem Stallmana, wspierała LMI20.

Z początku Stallman zaakceptował dążenie obu firm do komercjalizacji Lisp Machine, nawet gdyby miało to oznaczać więcej pracy dla niego. Obie firmy korzystały z licencji i kodu źródłowego Lisp Machine OS MIT i jego zadaniem było modyfikowanie tego kodu, aby dotrzymać kroku wszystkim zmianom wprowadzanym w firmach. Choć umowa Symbolics z MIT dawała mu prawo do przeglądania, ale nie do kopiowania kodu źródłowego, nadal "pożyczał" atrakcyjne wyimki na zasadzie "gentleman's agreement" i tradycyjnych zasad hakerskiego postępowania.

Jednakże 16 marca 1982 roku - zapamiętał tę datę dokładnie, gdyż był to jego urodziny - zarząd Symbolics zerwał dwustronną umowę. Był to ruch strategiczny. LMI, główny rywal na rynku maszyn lispowych, w znacznej mierze opierał działanie na kopii kodu źródłowego AI Lab Lisp Machine.

Zamiast wspierać rywala, zarząd Symbolics zdecydował się na ściślejsze przestrzeganie zasad zapisanych w umowie. Jeżeli AI

Lab chciało korzystać z osiągnięć Symbolics, powinno zainstalować system tej firmy i odciąć dostęp LMI do kodu źródłowego.

Rozdrażniło to Stallmana jako osobę odpowiedzialną za uaktualnianie laboratoryjnego kodu Lisp Machine. Traktując to oświadczenie jako "ultimatum", zemścił się, odłączając mikrofalowe połączenie Symbolics z AI Lab. Przysiągł sobie nigdy nie dotykać kodu Symbolics i pozostać wierny współpracy z LMI. "Uważałem, że AI Lab jest tu krajem neutralnym, jak Belgia podczas I wojny światowej - twierdzi. - Jeżeli ktoś dokonuje inwazji na Belgię, jej rząd jest zmuszony ogłosić stan wojny z napastnikiem i sprzymierzyć się z Wielką Brytanią i Francją".

Przebieg tej wojny "symbolicznej" ("Symbolics" War), trwającej w latach 1982-1983, zależy od źródła danych, czyli od tego, kto o niej opowiada. Gdy zarząd Symbolics zauważył, że zmiany wprowadzone przez jego firmę wciąż ukazują się w aktualizacjach AI Lab Lisp Machine, zdecydował się na zainstalowanie "szpiega" na terminalu Stallmana. Stallman twierdził, że wszystkie funkcje pisał od nowa, korzystając ze zgodnego z umową prawa przeglądania kodu i znosząc dodatkowe uciążliwości związane z koniecznością uczynienia kodu jak najmniej podobnym do oryginału. Zarząd Symbolics twierdził co innego i skierował zażalenie do administracji MIT. Zgodnie z wydaną w roku 1994 książką Harveya Newquista *The Brain Makers: Genius, Ego, and Greed, and the Quest for Machines That Think*²¹, administracja ostrzegła Stallmana, by "trzymał się z daleka" od projektu Lisp Machine²². Stallman twierdzi, że administracja go wsparła. "Niczym mnie nie straszono - mówi. - Robiłem zmiany jak dawniej. Aby się dodatkowo zabezpieczyć, przestałem czytać ich kod źródłowy. Czytałem tylko dokumentację i na jej podstawie wprowadzałem zmiany".

Niezależnie od wyniku kłótni utwierdził a Stallmana w jego postanowieniu. Bez kodu źródłowego uzupełniał braki systemu

zgodnie z własnymi przekonaniem i smakiem oraz nakłonił innych w AI Lab do informowania go o wszelkich niedociągnięciach i błędach maszyny lispowej. Programistom z LMI zapewnił bezpośredni dostęp do wszystkich wprowadzanych zmian. "Zamierzał emukarać Symbolics, nawet gdyby to miał abyć ostatnia rzecz, jaką zrobiłem" - twierdzi. To ważne stwierdzenie. Nie tylko rzuca światła na niepokorną naturę Stallmana, lecz również ujawnia poziom emocji, które towarzyszył y tej "wojnie". Zgodnie z inną historią opowiedzianą przez Newquista, Stallman był w pewnym momencie tak zirytowany, że wysłał e-mail: "Owinę się pasem z dynamitem i złożę wizytę w biurze Symbolics²³". Choć Stallman twierdzi, że nie pamięta tego e-maila i że opowiadanie o nim jest jedynie zjadliwą złośliwością, przyznaje, że takie myśli krążyły mu po głowie. "Pamiętam fantastyczne pomysły zniszczenia siedziby Symbolics i legnięcia pod jej gruzami. Na szczęście moje życie był o dla mnie ważniejsze"²⁴.

Jego rozpacz był a wielka, gdyż jednocześnie obserwował zniszczenie tego, co uważał za "swoją dom" - zwartej hakerskiej subkultury AI Lab. W e-mailowym wywiadzie udzielonym później Levy' mu ukazał siebie jako postać podobną do historycznego Ishi, ostatniego żyjącego członka Yahi, plemienia indiańskiego z północno-zachodniego wybrzeża Pacyfiku, które zostało wygubione podczas walk z Indianami w latach sześćdziesiątych i siedemdziesiątych XIX w. Ta analogia pozwalał a opisywać jego przetrwanie w epickiej, niemal mitologicznej formie. W rzeczywistości był to jedynie przypisek do historii napięć powstających między nim i resztą hakerskiej społeczności AI Lab tuż przed rozł aniem Symbolics-LMI. Wielu z jego kolegów widziało Symbolics nie jako zł owieszczą siłę niszczącą, lecz raczej jako zbyt późno pojawiającą się alternatywną ofertę. Firma komercjalizująca Lisp Machine wygnał a projektowanie oprogramowania z bezpiecznej izolacji, jaką było AI Lab, na otwarty komercyjny rynek, gdzie menadżerowie ustanawiali wł aśne prawa. Niektórzy uważali

Stallmana nie za bojownika trwającego na ważnym posterunku, lecz za przeżytek minionej epoki.

Stallman nie wdaje się w dyskusje na temat alternatywnego widzenia tamtych zdarzeń. Jak twierdzi, była to kolejna, po postawieniu "ultimatum", przyczyna nasilająca jego wrogość do Symbolics. Wspomina, że jeszcze przed zatrudnieniem przez Symbolics większości hakerów z AI Lab ci, którzy później tam przeszli, unikali go. "Już mnie nie zapraszano do Chinatown - wspomina. - Zgodnie ze zwyczajem wprowadzonym przez Greenblatta, jeżeli ktoś wybierał się na obiad gdzieś na zewnątrz, puszczał o tym informację, szukając chętnych do towarzystwa. Gdzieś w latach 1980-1981 przestano mnie zapraszać. Nie tylko mnie nie zapraszano, lecz jak mi później ktoś opowiedział, na niego i innych wywierano nacisk, by kłamali i ukrywali przede mną, że inni wybierają się do restauracji".

Czuł złość do hakerów zabawiających się tymi dziecinnymi postaciami ostracyzmu, zaś spór z Symbolics wywołał w nim wściekłość innego rodzaju, wściekłość tracącego dom na jego burzyciela. Gdy Symbolics wstrzymało wysyłanie zmian wprowadzanych w kodach, odpowiedział niemal ciągłym przesiadywaniem w AI Lab i tworzeniem kodów zmieniających funkcji od zera, jedynie na podstawie opisów. Jego frustracja zapewniła użytkownikom zewnętrznym Lisp Machine pełny otwarty dostęp do wszystkich zmian wprowadzanych przez Symbolics.

Utrwalił oto także wewnątrz hakerskiej społeczności legendę Stallmana. Znany już z pracy nad Emacsem, teraz pokazał, że potrafi dotrzymać kroku całemu zespołowi programistów pracujących w Symbolics, a w grupie tej znajdowało się kilku hakerów o legendarnej sławie. Było to jedno z największych jednostkowych osiągnięć Wieku Informacji, a może innych wieków również. Steven Levy nazwał to "hakerskim majstersztykiem", a samego Stallmana określił jako "wirtualnego Johna Henry'ego". W swoich uwagach zapisał, że

członkowie rywalizującej z nim grupy pracującej dla Symbolics, mogli mieć pretensję do swego byłego kompana, ale musieli czuć dla niego respekt. Levy cytuje Billa Gospera, hakera, który pracował dla Symbolics w oddziale w Palo Alto. Wypowiedź jest pełna podziwu dla pracy wykonywanej w tym czasie przez Stallmana:

Gdy w tym, co zrobił Stallman, znajdowałem coś, co mogłem uznać za źle wykonane (a właściwie ktoś inny usilnie mnie przekonywał, że jest to źle wykonane), mogłem zawsze powiedzieć: "Poczekaj, on pracując po nocach, nie ma z kim tego omówić. Pracuje samotnie! To niemożliwe, aby taką pracę wykonywał jeden człowiek!"²⁵

Dla Stallmana miesiące spędzone na dotrzymywaniu kroku zespołowi Symbolics są źródłem dumy i goryczy. Jako zdeklarowany liberał, którego ojciec walczył w II wojnie światowej, Stallman nie jest pacyfistą. Pod wieloma względami wojna z Symbolics stał się rytualnym przejściem tam, dokąd zmierzał od czasu przyłączenia się do grupy zespołu AI Lab przed dziesięcioma laty. Niestety, zbiegło się to z bolesnym rozpadem hakerskiej subkultury, która kształtowała jego naturę od wieku młodości. Pewnego dnia, gdy oderwał się od pisania kodu i przechodził przez biuro, natknął się na stojącą w kącie, już nieużywaną maszynę PDP-10. Był to dla niego traumatyczne przeżycie. Patrzenie na zamarłe lampki, które niegdyś błyskaniem informowały o stanie wewnętrznego oprogramowania, było dla niego przeżyciem podobnym do oglądania dobrze zakonserwowanych zwłok ukochanej osoby. "Widząc tę maszynę martwą, zacząłem płakać - wspomina. - Stojąca w kącie i przez wszystkich zapomniana, uświadomiła mi, jak dalece nasza społeczność została zniszczona".

Miał niewiele okazji do narzekania. Lisp Machine, pomimo całej wrzawy i wściekłości, jakie wywołała, pomimo wysiłku, jakiego wymagał o jej stworzenie, istniał jedynie gdzieś na skraju pola walki nowych rynkowych technologii. Niepowstrzymany bieg ku miniaturyzacji niósł nowe potężne mikroprocesory, a ich użycie

stworzył onowy świat sprzętu komputerowego i oprogramowania, który zaczął pochłaniać rozwiązania starszej generacji, jak rozrastająca się metropolia pochłania stare pustoszące wioseczki rozsiane w jej okolicy.

Na szczycie tej mikroprocesorowej fali napłynęły setki i tysiące nowych komercyjnych programów, z których każdy miał przypiętą metkę informującą o ograniczeniach licencyjnych uniemożliwiających hakerom przeglądanie, zmienianie i przekazywanie sobie kodów źródłowych. Na początku licencje były "grubo ciosane" i źle napisane, ale do roku 1983 zostały tak poprawione, że zaspokajały wymagania sądów i odstraszały potencjalnych intruzów. Oprogramowanie, które koncerny elektroniczne niegdyś uważał yjedyne za dodatek do wytwarzanego przez nie kosztownego sprzętu, za dodatkową atrakcję, rodzaj deseru mającego przyciągnąć klientów, stało się teraz daniem głównym. Żadni nowych gier i funkcji użytkownicy wyzbyli się dawnego obyczaju analizowania przepisu po spożyciu posiłku.

Nigdzie nie było to bardziej widoczne niż w dziedzinie systemów operacyjnych komputerów osobistych; Apple Computer i Commodore zarabiał ymiliony, sprzedając maszyny z wbudowanymi systemami operacyjnymi. Większość użytkowników nie protestował a, nie otrzymując przy zakupie plików z kodami źródłowymi, gdyż nie zdawała sobie sprawy z zasad hakerskiej kultury i niechęci tego środowiska do oprogramowania sprzedawanego wyłącznie w postaci binarnej. Niewielu anarchistycznych zwolenników hakerskiej etyki starało się ją upowszechniać w nowym rynkowym świecie, ale większość z tego zrezygnował a, gdyż rynek wynagradzał tych, którzy szybko pisali nowe programy i wykazywali zmysł handlowy nakazujący opatrywanie wyników pracy klauzulami ochrony praw autorskich.

Prym w tej dziedzinie wiodł Bill Gates, o dwa lata młodszy od Stallmana absolwent Harvardu. Stallman, wysyłając swoje przesłanie do grupy dyskusyjnej net.unix-wizards, nie wiedział ,

że przed siedmiu laty Gates, wówczas początkujący przedsiębiorca i główny udziałowiec założonej w Albuquerque firmy programistycznej Micro-Soft, później zmienione na Microsoft, wystosował swoją własną odezwę do posiadaczy komputerów osobistych. Była to odpowiedź skierowana do użytkowników PC-tów kopiujących oprogramowanie tworzone przez Micro-Soft. List otwarty do hobbystów Gatesa odzierał ze złudzeń społeczny ruch twórców oprogramowania.

"Kto w pocie czoła zechce wykonywać profesjonalną pracę za nic? - pytał Gates. - Jaki hobbysta poświęci trzy lata na programowanie, wynajdywanie błędów, tworzenie dokumentacji po to, by rezultat oddać innym za darmo?"²⁶

Niewielu hakerów z AI Lab czytało przesłanie Gatesa z 1976 roku, ale i bez tego byli świadomi odmiennego nastawienia do oprogramowania firm komercyjnych i nastawionych komercyjnie twórców oprogramowania. Dlaczego traktować oprogramowanie jako coś, co nic nie kosztuje, skoro rynek mówi inaczej? Lata siedemdziesiąte były wstępem do lat osiemdziesiątych i sprzedaż oprogramowania stawała się coraz częściej czymś więcej niż zwykle odzyskiwanie poniesionych kosztów; zaczął się przerażać w manifest polityczny. W czasach, gdy administracja Reagana przystąpiła do demontażu federalnych regulacji prawnych i programów wsparcia finansowego - będących spadkiem jeszcze po czasach wielkiego kryzysu - część hakerów uznała wyzute z konkurencyjności zasady hakerskiej etyki za antyamerykańskie. W najlepszym razie zaczęto je traktować jako antykorporacyjny balast z epoki późnych lat sześćdziesiątych i lat siedemdziesiątych. Jak bankier z Wall Street, znajdujący w szafie z dwurzędowymi garniturami starą koszulę w ciapki i bez kołnierzyka, wstydliwie ukrytą wśród nowych koszul z kołnierzykami i francuskimi mankietami, tak wielu programistów z zakłopotaniem wspominało zasady hakerskiej etyki jako echo dawnej idealistycznej epoki.

Dla człowieka, który przez całą dekadę lat sześćdziesiątych

stanowił kłopotliwe wspomnienie lat pięćdziesiątych, nie miał o znaczenia niedotrzymywanie kroku rówieśnikom. Jako programista przywykł ydo pracy na najlepszym sprzęcie i oprogramowaniu, stanął przed "trudnym wyborem moralnym": albo porzuci swoje ideowe zastrzeżenia dotyczące oprogramowania "właściwościowego", jak on i jego koledzy-hakerzy określali wszelkie oprogramowanie opatrzone klauzulą prywatnych praw autorskich lub ograniczającą prawo kopiowania i modyfikowania kodu licencją końcowego użytkownika, albo poświęci się budowaniu alternatywnego systemu operacyjnego i oprogramowania użytkowego rozpowszechnianego bez ograniczających praw własności. Po ostatnich miesiącach bolesnych doświadczeń walki z Symbolics miał większą ochotę na drugie rozwiązanie. "Zakładam, że mógłbym w ogóle przestać zajmować się komputerami i pomimo braku zdolności w tym kierunku, zostać kelnerem. Może nie w modnej restauracji, ale gdzieś by mnie chyba zatrudnili." Zostanie kelnerem i porzucenie na stałe oprogramowania, które przyniosło mu tyle przyjemności, mogło od niego oznaczać rezygnację z wszelkiej aktywności. Patrząc wstecz na swe życie od czasu przeprowadzki do Cambridge, mógł łatwo rozpoznać długie okresy, w których pisanie programów było czystą przyjemnością. Zatem zamiast je porzucić postanowił zająć się nim jeszcze intensywniej.

Jako ateista odrzucał takie pojęcia jak los, dharma²⁷ i wezwanie Opatrzności do podjęcia zadania. Pomimo to nie sądził, aby decyzja odrzucenia oprogramowania z prawami własności i budowania systemu operacyjnego tylko w celu ułatwienia życia innym była czymś normalnym. To połączenie własnego uporu, zdolności przewidywania i mistrzostwa w pisaniu programów pchało go na ścieżkę zbaczającą z głównej drogi, ścieżkę, której istnienia inni nawet nie zauważali. W roku 1999 w rozdziale książki "Open Sources"²⁸, wyjaśniając motywy decyzji, powołuje się na słowa żydowskiego mędrca Hillela:

Jeżeli nie istnieję dla samego siebie, któż istnieć będzie dla mnie?

Jeżeli istnieję tylko dla siebie, kimże jestem?

Jeżeli nie teraz, to kiedy?²⁹

If I am not for myself, who will be for me?

If I am only for myself, what am I?

If not now, when?³⁰

Przemawiając publicznie, unika odwołań religijnych i wyjaśnia swą decyzję w sposób bardziej pragmatyczny. "Spytał em siebie: w jaki sposób ja, twórca systemów operacyjnych, mogę poprawić sytuację? Po rozważaniu sprawy doszedłem do wniosku, że właśnie ktoś tworzący systemy operacyjne jest osobą najbardziej pożądaną do rozwiązania tego problemu". Gdy ostatecznie podjął decyzję, wszystko inne znalazło swe właściwe miejsce. Mógł powstrzymać się od korzystania z programów, które zmuszały go do kompromisów etycznych, i poświęcić życie tworzeniu oprogramowania, które miało ułatwić innym pójście w jego ślady. Przyrzekł sz, że stworzy ogólnodostępny system operacyjny lub, oczywiście w późnym wieku, "umrze pracując nad nim", w styczniu 1984 roku zrezygnował z pracy w MIT i przystąpił do tworzenia GNU. Rezygnacja pozbawiła go wsparcia MIT, ale wciąż miał wystarczającą liczbę przyjaciół w AI Lab, by zapewnić sobie wolny dostęp do pracowni uczelni. Mógł także liczyć na czasowe zatrudnienie w roli zewnętrznego konsultanta, co stanowiło wsparcie we wczesnym okresie projektu GNU. Rezygnacja z pracy w MIT zakończyła wszystkie dyskusje na temat konfliktu interesów i praw własności tworzonego oprogramowania. Człowiek, którego strach przed społeczną izolacją w okresie dorastania wciskał coraz głębiej w środowisko AI Lab, teraz postanowił odciąć się od niego murem. Przez kilka pierwszych miesięcy pracy odizolował się także od

społeczności uniksowej, choć jego obwieszczenie wysłał do grupy net.unix-wizards wywołało sympatyczny odzew i kilku wolontariuszy zgłosił o chęć udziału w krucjacie już we wczesnym jej stadium.

"Reakcja społeczności była dość jednolita - wspomina Rich Morin, w tym czasie lider grupy użytkowników Unixa. - Ludzie mówili: 'To wspaniały pomysł. Pokaż nam kod. Pokaż, że to można zrobić'."

Jako rasowy haker Stallman zaczął od poszukiwań istniejących programów i narzędzi, które nadawałyby się do przekształcenia w programy i narzędzia GNU. Na samym początku zwrócił uwagę na kompilator o nazwie VUCK, przekształcający programy napisane w języku C w kod czytelny dla maszyny. W tłumaczeniu z holenderskiego akronim nazwy oznaczał Free University Compiler Kit31. Pełen optymizmu zapytał autora programu, czy zestaw jest wolnym oprogramowaniem. Został poinformowany, że "Free University" oznacza Vrije Universiteit in Amsterdam, co bardzo go rozgoryczyło.

"Odpowiedział sztywno, że uniwersytet jest wolny, ale jego oprogramowanie nie - wspomina Stallman. - Tak więc zdecydowałem, że pierwszym programem, który napiszę dla projektu GNU, będzie wielojęzyczny i działający na wielu platformach kompilator".

Ostatecznie znalazł kompilator języka Pascal napisany przez programistów z Lawrence Livermore National Lab. Zgodnie z jego ówczesną wiedzą, kompilator był oprogramowaniem wolnym i mógł być swobodnie kopiowany i modyfikowany. Niestety, miał jedną istotną wadę: każdy program w całości zapamiętywał w pamięci operacyjnej, zmniejszając obszar dostępny dla innych działań. To co dla wielkich komputerów mainframe było dopuszczalne, dla Uniksa stało się paraliżującą barierą, gdyż maszyny sterowane tym systemem były za małe i miały za małą pamięć do całościowej obsługi generowanych plików. Istotnym postępem był odobudowanie przez niego nakładki/interfejsu zgodnego z językiem C. Jednakże przed

nadejściem lata zdecydował się na napisanie nowego kompilatora od podstaw.

We wrześniu 1984 roku uznał napisanie kompilatora za zadanie do wykonania w najbliższym czasie i rozpoczął poszukiwania narzędzi niskiego poziomu. Zaczął także rozwijać wersję GNU Emacsa, programu, którego rozwój nadzorował przez dekadę. Była to decyzja strategiczna. Społeczność uniksowa używała w tym czasie dwóch edytorów; jednym był vi napisany przez Sun Microsystems wraz z Billem Joyem, a drugim ed, stworzony przez pracującego dla Bell Labs naukowca Kena Thompsona (współtwórcę Uniksa). Oba były przydatne i popularne, ale żaden nie miał rozszerzalnej w nieskończoność natury Emacsa. Przygotowując Emacsa dla użytkowników Uniksa, Stallman miał doskonałą okazję do pokazania swej programistycznej biegłości. Była to także okazja przystosowania użytkowników Emacsa do stallmanowskiego sposobu myślenia.

Patrząc wstecz, twierdzi że wówczas nie widział strategicznych aspektów swojej decyzji. "Chciał mieć Emacsa i był a po temu doskonała okazja" - powiada.

Było to częściowo ponowne "odkrywanie Ameryki", przynajmniej tak to odczuwała jego hakerska wrażliwość. Pisząc uniksową wersję Emacsa, wkrótce wkroczył na ścieżkę, którą przed nim wydeptał James Gosling, student starszych lat w Carnegie Mellon, autor opartej na języku C wersji Emacsa, nazwanej GOSMACS. Wersja Goslinga zawierała interpreter korzystający z uproszczonej wersji języka Lisp, zwanej MOCKLISP.

Zdecydowany na zbudowanie GNU Emacsa na podobnych lispowych podstawach, Stallman korzystał obficie z innowacji Goslinga. Gosling opatrzył GOSMACS-a klauzulą copyrightu i sprzedał prawa prywatnej firmie UniPress, ale Stallman cytował współtwórcę programu, który twierdził, że Gosling, jeszcze jako doktorant w Carnegie Mellon, zapewniał współpracowników, że ich praca pozostanie dostępna dla wszystkich. Jednakże UniPress, dowiedziawszy się o projekcie Stallmana, zaczął

straszyć egzekwowaniem copyrightu. Stallman ponownie stanął przed koniecznością pisania wszystkiego od podstaw.

Stosując inżynierię wstecz³², mógł na podstawie interpretatora Goslinga stworzyć w pełni funkcjonalny interpreter Lispu i ominąć sporny punkt używania interpretera Goslinga. Niemniej jednak rozjątrzył go bardzo sam pomysł sprzedaży przez twórców oprogramowania praw do niego. Podczas przemówienia w Szwedzkim Królewskim Instytucie Technologicznym w roku 1986 przytaczał zdarzenie z UniPressem jako jeszcze jeden przykład niebezpieczeństw dotyczących oprogramowania z ograniczonymi prawami własności.

"Czasami myślę, że najlepszą rzeczą, jakiej mógłbym w życiu dokonać, byłoby zebranie w jeden stos całego oprogramowania z prywatnymi prawami własności i zawartymi tam tajemnicami handlowymi i rozdawanie kopii na rogu ulicy wszystkim chętnym. W ten sposób tajemnice przestałyby być tajemnicami - twierdzi. - Być może byłby to o wiele efektywniejszy sposób udostępnienia ludziom wolnego oprogramowania niż pisanie go od podstaw; ale wszyscy jesteśmy na to zbyt tchórzliwi".

Pomimo stresu, jaki wywoływał a, dysputa na temat innowacji Goslinga miała towarzyszyć Stallmanowi i ruchowi wolnego oprogramowania przez dłuższy czas. Zmusiła Stallmana do rozpoznania słabości społeczności Emacs i systemu wzajemnego zaufania, który zezwalał na powstawanie odgałęzień o problematycznej wartości. Skłonił go także do wyraźniejszego sformułowania celów politycznych ruchu wolnego oprogramowania. Po udostępnieniu GNU Emacsa w roku 1995 wydał Manifest GNU (GNU Manifesto) będący rozszerzeniem oryginalnego przesłania z września 1983 roku. Włączył do dokumentu długi rozdział, w którym wyliczył argumenty przytaczane przez komercyjnych i akademickich programistów na usprawiedliwienie rozpowszechniania oprogramowania z prywatnymi prawami własności. Jeden argument: "Czyż programistom nie należy się nagroda za ich kreatywność" wywołał jego wściekłość spowodowaną ostatnim

epizodem z goslingową wersją Emacs'a:

"Jeżeli cokolwiek wymaga nagrody, to tylko działanie dla dobra ogółu - odpowiadał .- Może nim być również kreatywność, ale tylko wtedy (sic!), gdy społeczeństwo może swobodnie korzystać z jej owoców. Jeżeli programiści żądają nagrody za tworzenie innowacyjnych programów, powinni też żądać nakładania kar za ograniczanie możliwości korzystania z rezultatów swej pracy"³³.

Pojawienie się GNU Emacs oznaczało, że projekt GNU ma wreszcie do pokazania jakiś kod. Towarzyszył ytemu zwykłe trudności związane z przedsięwzięciem programistycznym. W miarę jak uniksowi twórcy zaczęli się interesować i zajmować nowym programem, zaczęły napływać pieniądze, podarunki i prośby o przesłanie taśm. Do zajmowania się biznesową stroną projektu GNU Stallman wciągnął kilku kolegów i stworzył Fundację Wolnego Oprogramowania (ang. Free Software Foundation - FSF), organizację pozarządową, której celem miał o być wspieranie projektu i umożliwianie mu szybszego osiągania celów. Ze Stallmanem jako prezesem i grupą hakerów w roli członków zarządu fundacja stała się oficjalnym przedstawicielem projektu GNU.

Robert Chassell, pracujący wówczas w Lisp Machines, Inc., w wyniku rozmowy ze Stallmanem podczas obiadu został jednym z pięciu członków zarządu. Został także skarbnikiem organizacji, która to rola, z początku niewielka, zaczęła szybko nabierać większego znaczenia.

Wspomina: "W roku 1985 nasze całkowite przychody i wydatki zamykały się w kwocie 23 000 dolarów. Richard miał swoje biuro, z którego my również korzystaliśmy. Wszystkie nasze dobra, a zwłaszcza taśmy upychałyśmy pod biurkiem. Tak było do czasu, aż LMI użyczyło nam na kredyt pewnej przestrzeni do magazynowania tego, co posiadaliśmy, łącznie z taśmami."

Fundacja Wolnego Oprogramowania poza rolą oficjalnego przedstawicielstwa projektu zaczęła również stanowić punkt przyciągający wszystkich rozczarowanych programistów. Rynek

unikswy, choć w początkowym okresie projektu GNU jeszcze dość otwarty, zaczynał się już stawać coraz bardziej konkurencyjny. Starając się zwiększyć uzależnienie klientów, firmy zaczynały ograniczać ich dostęp do kodów źródłowych, co z kolei przyspieszało o liczbę poszukujących oprogramowania dostarczanego przez projekt GNU. Wielcy magowie Uniksa, którzy niegdyś uważali Stallmana za krzykliwego przygłupa, ogłosili go teraz programistyczną Kasandrą.

"Wielu ludzi, póki im się to nie przytrafiło, nie uświadamiało sobie, jak frustrujące może być odebranie programu, nad którym się pracował latami - mówi Chassell, podsumowując odczucia i opinie wielu piszących w ciągu lat do Fundacji Wolnego Oprogramowania. - Gdy się to stanie, zaczynasz mówić do siebie 'Zaraz, zaraz, poczekajcie'".

Do decyzji Chassella o udziale w ruchu wolnego oprogramowania przyczynił się osobista strata. Przed zatrudnieniem się w LMI jako wynajęty pracownik napisał książkę wprowadzającą do Unixa dla Cadmus, Inc., firmy programistycznej z Cambridge. Gdy firma upadła, grzebiąc wraz z sobą prawa autorskie do książki, Chassell chciał te prawa odkupić, ale bez rezultatu.

"Ta książka wciąż tkwi na jakiejś półce, bezużyteczna, niedostępna do skopiowania, pochłonięta przez system - mówi Chassell. - O ile jestem w stanie ocenić własną pracę, mogę powiedzieć, że jest to całkiem niezły wstęp. Przez jakieś trzy lub cztery miesiące można by było go przekształcić we wprowadzenie do dzisiejszego GNU/Linuxa. Całe zdobyte doświadczenie, oprócz tego, co pozostało w mojej pamięci, zostało stracone".

Zmuszony do oglądania swej pracy ginącej bez śladu, gdy jego poprzedni pracodawca walczył ze zbliżającym się bankructwem, Chassell odczuł na własnej skórze, że wściekłość mogła doprowadzić Stallmana niemal do apopleksji. "Stał się dla mnie jasne, że jeżeli chcę przyzwoicie przeżyć życie, nie mogę pozwalać na przetrzymywanie żadnej jego części w zamknięciu

- mówi Chassell. - Rzecz w tym, by zachować wolność wprowadzania zmian i poprawek, gdy chcemy to robić. Daje to poczucie radości, że coś, co tworzyliśmy przez kilka lat życia, ma jakąś wartość. Bez tego mamy wrażenie, że coś zostało nam odebrane i wyrzucone, a przynajmniej, że nie mamy już z tym żadnego kontaktu. Jest to utrata części życia".

Przypisy

1 Usenet to ogólnosiwiatowy komputerowy system dyskusyjny wykorzystujący do transmisji danych Internet oraz inne sieci komputerowe. Wymiana informacji odbywa się w ponad 30 000 grup dyskusyjnych, zawierających teksty rozpoczynające dyskusje, tzw. artykuły, oraz komentarze do tychże artykułów, tzw. kontynuacje wątków. Uniksowe grupy dyskusyjne usenetu są odwiedzane codziennie przez około 15 milionów ludzi z różnych krajów

- przyp. tłum.

2 Skrót od Thanksgiving Day (Dzień Dziękczynienia) - czwarty czwartek listopada w USA, drugi poniedziałek października w Kanadzie - święto obchodzone na pamiątkę pierwszego dziękczynienia członków kolonii Plymouth w 1621 r., kiedy pielgrzymi przybyli na żaglowcu Mayflower. Po pierwszych udanych zbiorach dziękowali za nie Opatrzności. Istnieją również inne wersje pochodzenia tego święta (patrz: np. <http://www.stos.opoka.org.pl/polus/thanksgv.htm>) - przyp. tłum.

3 Patrz: Richard Stallman Initial GNU Announcement (wrzesień 1983).

4 Patrz rozdział 4. str...

5 Stacje robocze budowane do późnych lat osiemdziesiątych przez Symbolics, Inc. we współpracy z MIT AI Lab były wyposażone w pisany w Lispie system operacyjny. Właściwie wszystko: narzędzia systemu operacyjnego, aplikacje i dane były tam traktowane jako obiekty Lispu, sterowane poleceniami

pisany w tym języku. Zapewniało to wysoką integrację i ułatwiało wykonywanie wszelkich operacji i zmian, ale stanowiło wielkie niebezpieczeństwo w przypadku krakerskiego włamania - przyp. tłum.

6 DEC (ang. Digital Equipment Corporation) - założona w 1957 r. przez K. Olsena firma komputerowa. Obecnie jest jednym z czołowych amerykańskich producentów sprzętu komputerowego i oprogramowania. Najbardziej znane produkty to. słynna w latach 70-tych seria minikomputerów PDP oraz komputery z serii VAX (od stacji roboczych do superkomputerów VAX 9000) - przyp. tłum.

7 AT&T - wielka amerykańska firma telekomunikacyjna sięgająca czasów wynalazcy telefonu Alexandra Grahama Bella. Więcej informacji można znaleźć pod adresem: <http://www.att.com/history> - przyp. tłum.

8 Komputer główny (ang. mainframe computer) - komputer o dużej mocy obliczeniowych obsługujący za pomocą terminali wielu użytkowników - przyp. tłum.

9 Patrz: Marshall Kirk McKusick "Twenty Years of Berkeley Unix", Open Sources (O'Reilly & Associates, Inc., 1999) s. 38.

10 Patrz: Richard Stallman (1986).

11 Informacje o tej innowacyjnej firmie, mającej znaczny udział w utworzeniu między innymi ARPANET-u, poprzednika Internetu, można znaleźć pod adresem <http://www.bbn.com> - przyp. tłum.

12 Liczne źródła: patrz: wywiad z Richardem Stallmanem, email Geralda Sussman, i Jargon File 3.0.0. <http://www.clueless.com/jargon3.0.0/TWENEX.html>.

13 Top - wierzchołek, szczyt (tops - l.m.) - przyp. tłum.

14 Patrz: http://www.as.cmu.edu/~geek/humor/See_Figure_1.txt

15 Patrz: "MIT AI Lab Tourist Policy".

16 Wykonywanie zadań/procesów na listach, które są tu odpowiednikiem zbiorów rekordów danych (baz danych) -

przyp. tł um.

17 DARPA, Defence Advanced Research Projects Agency, agencja Departamentu Obrony USA do spraw zaawansowanych technicznie programów badawczych. Zarządza badaniami o tematyce obronnej. Znana m.in. ze sponsorowania prekursorskich prac w dziedzinie sieci komputerowych. Z jej inicjatywy powstał a w 1968 pierwsza sieć rozległa - ARPANET - przyp. tł um. wg encyklopedii internetowej <http://wiem.onet.pl>

18 Patrz: przypis 11. w rozdziale 6. - przyp. tł um.

19 W częściowym skrócie Lisp Machine OS - system operacyjny maszyny lispowej - przyp. tłum.

20 H.P. Newquist, The Brain Makers: Genius, Ego, and Greed in the Quest for Machines that Think (Sams Publishing, 1994) s. 172.

21 Tworzące mózg: geniusz, ego i pożądanie oraz poszukiwanie myślącej maszyny - przyp. tłum.

22 H.P. Newquist, The Brain Makers: Genius, Ego, and Greed in the Quest for Machines that Think (Sams Publishing, 1994) s. 196.

23 Ibid. Newquist twierdzi, że o prawdziwości tej anegdoty przekonywało go kilku członków zarządu Symbolics. Twierdzili, że "e-mail wywołał falę podniecenia i spekulacji wśród pracowników, ale ostatecznie nikt nie potraktował tego poważnie".

24 http://www.as.cmu.edu/~geek/humor/See_Figure_1.txt.

25 Steven Levy, Hackers (Penguin USA [paperback], 1984): 426.

26 Bill Gates, An Open Letter to Hobbyists (3 lutego 1976).

27 Dharma - w hinduizmie norma określająca działania ludzkie, tak by były one zgodne z kosmicznym ła dem: rytą; w buddyzmie jeden z tzw. trzech klejnotów buddyzmu - nauka Buddy; w filozofii buddyjskiej - podstawowy element świata zjawisk, substrat poszczególnych momentów potoku doznań zmysłowych, uczuciowych, świadomościowych itp. - przyp. tł um. na podstawie encyklopedii internetowej

<http://wiem.onet.pl/>

28 Książka wydana przez wydawnictwo O' Rilly - patrz: następny przypis. Informacje na jej temat można znaleźć pod adresem <http://www.oreilly.com/catalog/opensources/>.

Stallman jest autorem jednego rozdziału. - przyp. tłum.

29 Tłumaczenie ze strony WWW: <http://arianiel.webpark.pl/> - przyp. tłum.

30 Richard Stallman, Open Sources (O'Reilly & Associates, Inc., 1999) s 56. Do tego stwierdzenia Stallman dodał przypis: "Jako ateista nie postępuję według zaleceń przywódców religijnych, ale czasami podziwiam to, co mówili".

31 Można to tłumaczyć dwojako, jako "wolny uniwersytecki zestaw kompilacyjny" lub "zestaw kompilacyjny wolnego uniwersytetu" - przyp. tłum.

32 Reverse Engineering (inżynieria wsteczna albo inżynieria wstecz) - to określenie technologii wytwarzania produktu na podstawie jego cech. Na przykład urządzenia mechaniczne wymagają fizycznych pomiarów, w tym pomiarów wielu powtarzających się części dla poznania statystycznej tolerancji w oryginalnym projekcie. Wymagają one badań strukturalnych, żeby poznać odkształcenia przy obciążeniu, testów chemicznych dla dokładnego określenia materiałów, z jakich został zrobiony, testów tarcia dla określenia tempa zużycia itp. Podobnie, produkty elektroniczne mają całą serię badań, które prowadzą do określenia ich charakterystyki. W przypadku programowania celem inżynierii wstecz jest odtworzenie kodu źródłowego na podstawie istniejącego gotowego programu. Inżynierię wstecz nazywa się czasem inżynierią odtwarzania - przyp. tłum.

33 Richard Stallman, The GNU Manifesto (1985).
<http://www.gnu.org/manifesto.html>.

8. ŚW. IGNUCY

Maui High Performance Computing Center¹ mieści się w

jednopiętrowym budynku na zapylonych czerwonych wzgórzach nad miastem Kihei. Otoczone przez warte miliony dolarów krajobrazy i ekskluzywną posiadłość Silversword Golf Course², centrum wygląda na ośrodek naukowy zbudowany tu jedynie w celu marnotrawienia pieniędzy. Dalekie od pudełkowej i sterylnej ciasnoty Tech Square, a nawet rozciągniętych szeroko naukowych metropolii Argonne³ w Illinois i Los Alamos⁴ w Nowym Meksyku - MHPCC wygląda jak miejsce, w którym naukowcy więcej czasu poświęcają na opalanie się niż na swoje postdoktoranckie projekty badawcze. Jest to prawda tylko w połowie. Choć naukowcy z MHPCC rzeczywiście korzystają z możliwości rekreacji, pracę swą traktują poważnie. Według serwisu internetowego Top500.org śledzącego działanie najpotężniejszych superkomputerów na świecie, superkomputer IBM SP Power3, działający w MHPCC i wykonujący 837 miliardów operacji zmiennoprzecinkowych na sekundę, jest jednym z 25 najpotężniejszych komputerów na świecie. Jako wspólna własność University of Hawaii i U.S. Air Force dzieli swoje cykle pracy między rozgrywanie wojskowych problemów obliczeniowych sił lotniczych i wyliczenia związane z pracami badawczymi w dziedzinie fizyki wysokich energii. MHPCC jest miejscem, gdzie wysoka kultura badań naukowych i inżynierii w spokojnej równowadze współistnieje z kulturą wysp hawajskich. Slogan ze strony internetowej z roku 2000 określił to jako "Obliczenia komputerowe w raju".

Nie jest to miejsce, w którym moglibyśmy spodziewać się znaleźć Richarda Stallmana, człowieka, który spoglądając przez biurowe okno na przepiękny Maui Channel, mruczał cierpko: "Za dużo słońca". Jednakże przybył tu jako emisariusz mający przekazać przesłanie z innego komputerowego raju, musiał więc pogodzić się z wystawieniem na tropikalne słońce swej bladej skóry hakera.

Gdy przyjechał tam, aby wysłuchać przemówienia Stallmana, sala konferencyjna już przed czasem była wypełniona. Podział audytorium według płci był nieco, choć

niewiele, lepszy niż w Nowym Jorku: 85% mężczyzn i 15% kobiet, z czego około połowa ubrana w spodnie koloru khaki i koszulki z golfowym logo. Resztę stanowili miejscowi z twarzami w ciemnym kolorze ochry, odziani w krzykliwe kwieciste koszule, tak popularne w tej części świata. Jedynym resztkowym wskaźnikiem ich statusu komputerowców był y gadżety: telefony komórkowe Nokia, Palm Piloty⁵ i laptopy Sony VAIO.

Nie trzeba dodawać, że Stallman stojący przed słuchaczami w niebieskim, jednobarwnym T-shircie, luźnych, brązowych poliestrowych spodniach i białych skarpetkach, odróżniał się od nich jak kciuk od reszty palców. Fluorescencyjne światła sali konferencyjnej podkreślał yniezdrowy kolor jego nieprzywykłej do słońca skóry. Jego broda i długie włosy wywołały by krople potu na najchłodniejszym hawajskim karku. Nawet wytatuowanie sobie na czole napisu "mainlander⁶" nie wyróżniałoby go bardziej wśród zebranych.

Gdy Stallman beczynn timer przechadzał się na przodzie sali, kilku słuchaczy, ubranych w T-shirty z logo Manui FreeBSD Users Group⁷ (MFUG), śpiesznie ustawiało kamerę i sprzęt audiowizualny. FreeBSD to wolne oprogramowanie będące odgałęzieniem Berkeley Software Distribution, poważanej w latach siedemdziesiątych akademickiej wersji Uniksa. Pod względem technicznym jest konkurentem systemu operacyjnego GNU/Linux. Nadal w hakerskim świecie wszystkie wystąpienia Stallmana są rejestrowane z zapalem archiwistów-amatorów Grateful Dead⁸. Członkowie MFUG kierującego lokalnym ruchem wolnego oprogramowania otrzymali zadanie utrwalenia wszystkich złotych okrucich mądrości RMS⁹ i przekazania ich braci programistycznej w Hamburgu, Bombaju i Nowosybirsku.

Analogia z Grateful Dead jest właściwa. Często opisując poczynania biznesowe zgodne z modelem wolnego oprogramowania, Stallman podaje jako przykład Grateful Dead. Rezygnując z ograniczania możliwości nagrywania przez fanów

koncertów na żywo, zespół ten stał się czymś więcej niż tylko grupą rockową. Stał się ośrodkiem odrębnej społeczności, wspólnoty wyznawców jego muzyki. Z upływem czasu wspólnota ta stała się tak wielka i tak oddana, że grupa zaczęła unikać kontraktów nagraniowych i poświęcił się jedynie objazdom i koncertom na żywo. W czasie ostatniego turnee w roku 1994 za bilety zebrano sumę 52 milionów dolarów. Z tym sukcesem mogłoby się mierzyć kilka firm programistycznych, ale wspólnotowy aspekt społeczności wolnego oprogramowania spowodował, że w drugiej połowie lat dziewięćdziesiątych część z nich zaczęła przyznawać, że publikowanie kodu źródłowego może być dobrym pomysłem. Mając nadzieję na stworzenie rzeszy lojalnych użytkowników, takie firmy jak IBM, Sun Microsystems i Hewlett Packard zaczęły przestrzegać formy, jeżeli nie ducha, stallmanowskiego przesłania. Uznając GPL¹⁰ za Magna Carta¹¹ technologii przemysłu informatycznego, redaktor kolumny programistycznej w ZDNet¹², Evan Leibovitch, opisał rosnący wpływ GNU i uznał, że to coś więcej niż okresowy trend. "Ten ruch społeczny pozwala użytkownikom ponownie decydować o przyszłych poczynaniach - pisze Leibovitch. - Tak jak Magna Carta dała prawa poddanym brytyjskim, tak GPL wzmacnia prawa i wolność konsumentów za pomocą oprogramowania udostępnianego użytkownikom"¹³.

Wspólnotowo-plemienny aspekt społeczności wolnego oprogramowania pomaga wyjaśnić, dlaczego 40 dziwaków programistów przybyło na konferencję, by posłuchać Stallmana, zamiast pracować nad swymi projektami z dziedziny fizyki lub surfować po sieci, tworząc znaczące raporty. Inaczej niż w trakcie przemówienia w Nowym Jorku, tym razem Stallman zaczął bez wstępu i bez przedstawienia się. Gdy członkowie FreeBSD wreszcie uporali się ze sprzętem, po prostu zrobił kilka kroków do przodu i zaczął mówić, tłumiając inne głosy w sali.

"W większości przypadków w rozważaniach, jakie prawa

powinny rządzić używaniem oprogramowania, uczestniczą ludzie z firm programistycznych, którzy rozważają problem z dogodnego dla nich punktu widzenia - powiedział , rozpoczynając wystąpienie. - Rozmyślają, jakie zasady należy narzucić wszystkim, by płacili im masę pieniędzy. Dobry los pozwolił mi w latach siedemdziesiątych należeć do społeczności programistów, która wspólnie korzystała z oprogramowania i użyczała go innym. I dlatego zawsze spoglądam na to samo zagadnienie z innej strony i pytam: jaki rodzaj praw może umożliwić stworzenie dobrego społeczeństwa, dobrego dla ludzi, którzy je stanowią? Z tego powodu dochodzę do zupełnie innych odpowiedzi."

Jak zawsze, przytoczył swą opowieść o drukarce laserowej, we właściwym momencie uzupełniając wypowiedź gestem wyciągniętej ręki wskazującej w tłum słuchaczy. Poświęcił także chwilę na wyjaśnienie nazwy GNU/Linux.

"Niektórzy pytają mnie: Po co tyle starań o zdobycie zaufania do tego systemu? Najważniejsze, że robota została zrobiona, a nie to, czy ludzie się o tym dowiedzą. Rada byłaby dobra, gdyby była prawdziwa, ale zadaniem nie było samo zbudowanie systemu, lecz jego rozpowszechnienie wśród użytkowników komputerów. Aby to było możliwe, musimy zrobić wszystko, aby wszelkie działania na komputerze można było wykonać za darmo, tzn. aby oprogramowanie było wolne.^{14 i 15}"

"Jest jeszcze wiele do zrobienia" - dodał .

Dla niektórych słuchaczy były to informacje dobrze znane, dla innych trąciły wiedzę tajemną. Gdy jeden ze słuchaczy w koszulkach golfowych zaczął przysypiać, Stallman zamilkł i poprosił , by ktoś przebudził usypiającego.

"Ktoś kiedyś powiedział , że mój głos jest tak kojący, że mógłbym zostać uzdrowicielem - powiedział , powodując krótki wybuch śmiechu. - Zgaduję, że oznacza to, iż mogę wam pomóc zapaść w błogi, dający odprężenie sen. Zapewne niektórzy z was tego potrzebują, więc nie powinienem wam w tym przeszkadzać. Jeżeli potrzeba wam snu, śpijcie".

Wystąpienie skończyło się dyskusją na temat patentów oprogramowania, tematu o coraz większym znaczeniu dla obu stron: firm programistycznych i społeczności wolnego oprogramowania. Koncepcja patentów, wymyślona i napisana dla świata realnego, okazała się nieporęczna i niedostosowana do kruchego świata technologii informatycznych, tak jak nie pasowała do działalności Napstera. Różnica między ochroną programu za pomocą copyrightu i praw patentowych jest subtelna ale znacząca. W przypadku copyrightu twórca programu może ograniczać kopiowanie kodu źródłowego, ale nie powielanie idei i funkcji programu. Inny twórca oprogramowania, jeżeli nie ma ochoty na używanie programu zgodnie z narzuconymi warunkami, może dokonać inżynierii wstecznej, tzn. skopiować funkcjonalność, pisząc kod od nowa. Takie powielanie pomysłów jest powszechnie stosowane w przemyśle programistycznym, gdzie firmy izolują grupy zajmujące się inżynierią wsteczną, aby zabezpieczyć się przed oskarżeniami o szpiegostwo przemysłowe i nieuczciwe hokus-pokus. W żargonie informatycznym takie postępowanie jest określane jako clean room engineering¹⁶.

Ochrona patentowa działa odmiennie. Według Biura Patentowego Stanów Zjednoczonych (U.S. Patent Office) firmy i osoby indywidualne mogą za pomocą patentów chronić innowacyjne algorytmy i zgłaszać swe skargi do publicznego rozpatrzenia. Teoretycznie pozwala to właścicielom patentów czerpać korzyści z udostępniania chronionej informacji, a ich prawa wyłączności obowiązują w okresie ograniczonym zwykle do 20 lat. Praktyczna wartość tej ochrony jest dość problematyczna, gdyż często działanie programu ujawnia tajemnicę pomysłu. W przeciwieństwie do copyrightu, patent umożliwia jego właścicielowi tamowanie niezależnego tworzenia oprogramowania opartego na tym samym pomysle lub o takiej samej funkcjonalności.

W przemyśle informatycznym, gdzie okres dwudziestoletni to cała rynkowa epoka, patenty mają znaczenie strategiczne. Tam

gdzie takie firmy jak Microsoft i Apple niegdyś toczyły wojnę o copyright oraz "wygląd i odczucie" (ang. look and feel) różnych technologii, dzisiejsze firmy internetowe starają się otaczać patentowym płótnem pojedyncze aplikacje i modele biznesowe, czego najlepszym przykładem jest próba Amazon.com, która w roku 2000 usiłowała opatentować proces dokonywania zakupu jednym kliknięciem (ang. one-click online shopping process). Jednakże dla większości firm patenty chroniące oprogramowanie są bronią defensywną, której użycie w licencjach wzajemnych, umowach "krzyżowych" itp. służy wyrównaniu sił i międzyfirmowemu odprężeniu, złagodzeniu wzajemnych napięć. Jednakże nadal w kilku istotnych przypadkach algorytmów komputerowego szyfrowania i tworzenia obrazów graficznych dostawcy oprogramowania z powodzeniem tą umią poczynania rywali.

Według Stallmana, praktyka chronienia oprogramowania za pomocą patentów dramatycznie wzmacnia potrzebę zwiększenia hakerskiej czujności i uwydatnia polityczne korzyści, jakie niesie istnienie konkurencyjnego wolnego oprogramowania. Wskazuje, że patenty programistyczne czynią niedostępnymi pewne regiony rynku i twierdzi, że dzięki konkurencyjnej wydajności i cenie wolne oprogramowanie, takie jak GNU/Linux i FreeBSD, zaczęło już zdobywać swoje miejsce, wypierając z rynku oprogramowanie z prywatnymi prawami własności. Niestety, wydajność i cena są dla ludzi ważniejsze niż podstawowa kwestia wolności użytkowników i twórców oprogramowania¹⁷.

"To nie brak zdolności nie pozwala nam tworzyć lepszego oprogramowania - mówił Stallman. - Nie możemy tego robić, bo nie mamy do tego prawa. Ktoś zakazał nam działać dla dobra społeczeństwa. Co się dzieje, gdy użytkownicy odkrywają spowodowane tym luki w wolnym oprogramowaniu? Jeżeli z tego, co głosi ruch zwolenników oprogramowania open source utkwił aim w pamięci obietnica tworzenia programów wydajnych i godnych zaufania, powiedzą: 'Nie dacie takiego

produktu, jaki obiecywaliście. Ten program nie jest tak funkcjonalny, ani tak niezawodny, jak obiecywaliście. Brak mu takiej to a takiej funkcji. Okłamaliście nas.[†] Jeżeli natomiast zgadzają się z naszą opinią, że wolność jest wartością samą w sobie, powiedzą: "Dlaczego ci ludzie ośmielają się pozbawiać nas tej funkcji programu? Dlaczego ośmielają się ograniczać naszą wolność?" I dzięki takim odpowiedziom możemy wytrwać pod gradem ciosów wymierzanych przez odłamki eksplodującego pocisku z patentami".

Tego rodzaju komentarze oczywiście powodują sporo zamieszania. Większość obrońców oprogramowania open source sprzeciwia się prawom patentowym jeszcze głośniejsze niż Stallman. Jednakże nikt nie podważa jego opinii, że adwokaci oprogramowania open source kładą nacisk głównie na korzyści użyteczne a nie polityczne. Zwykle, zamiast podkreślać polityczne znaczenie wolnego oprogramowania, mówią o integralności jego konstrukcji opartej na hakerskim modelu rozwoju. Cytując oceny równorzędnych programów, malują obraz GNU/Linuksa lub FreeBSD jako lepiej skonstruowanych, lepiej sprawdzonych i, co z tego wynika, dla przeciętnego użytkownika bardziej godnych zaufania.

Nie oznacza to, że pojęcie "oprogramowanie open source" nie niesie żadnych politycznych implikacji. Termin ten przez jego adwokatów jest używany z dwóch powodów. Po pierwsze, eliminuje zamieszanie wokół słowa "wolny", które biznes interpretuje zwykle jako "o zerowych kosztach". Po drugie, pozwala firmom na rozpatrywanie zagadnienia wolnego oprogramowania raczej z technicznego niż etycznego punktu widzenia. Eric Raymond, jeden z twórców Open Source Initiative¹⁸ i doskonały haker, poparł używanie terminu "oprogramowanie open source" w eseju z roku 1999 "Zamknij się i pokaż im kod" (ang. Shut Up and Show Them the Code), gdzie efektywnie podsumował frustracje płynące z kroczenia polityczną drogą Stallmana.

Retoryka RMS potrafi uwieść wielu z nas. My, hakerzy,

jestemy skłonni do rozmyślań idealistami, do których łatwo trafiają takie słowa jak "zasada", "wolność" i "prawa". Nawet jeżeli nie zgadzamy się ze szczegółami programu, chcemy, aby retoryka RMS odnosiła skutek, sądzymy, że powinna działać na ludzi, i dziwimy się, że w 95% nie odnosi skutku, gdyż trafia na myślących inaczej niż my.¹⁹

W wymienionych przez Raymonda 95% mieszczą menadżerowie biznesu, inwestorzy i niehakerscy użytkownicy komputerów, którzy z prostego porównania liczb odczytują tendencje rynkowe. Bez znalezienia drogi do przekonania tych ludzi hakerzy wraz ze swoją ideologią będą zawsze tkwić na marginesie społeczeństwa:

RMS, naciskając na dyskutowanie o prawach użytkowników komputerów, wysłał do nas niebezpiecznie atrakcyjne zaproszenie do powtarzania starych błędów. Powinniśmy je odrzucić, nie dlatego że opiera się na złych zasadach, lecz dlatego, że język, którego używa w zastosowaniu do oprogramowania, nikogo prócz nas nie przekonuje. Wręcz przeciwnie - ludzi spoza naszej kultury wprowadza w zakłopotanie i odpycha.²⁰

Patrząc na Stallmana przedstawiającego osobiście swą polityczną wizję, trudno zauważyć cokolwiek wprowadzającego w zakłopotanie lub odpychającego. Jego wygląd może się nie wydawać zachęcający, ale przesłanie jest logiczne. Jeżeli ktoś ze słuchaczy pyta, czy świadomie unikający zaznaczania praw własności twórcy wolnego oprogramowania nie tracą możliwości dotrzymywania kroku postępowi najnowszych technologii, odpowiada, używając określeń zgodnych z jego własną wizją. "Sądzę, że wolność jest ważniejsza od postępu technicznego - mówił w trakcie swojego wystąpienia na Hawajach. - Zawsze wybiorę wolny program mniej zaawansowany technicznie, a nie jego lepszy technicznie odpowiednik z zastrzeżonymi prawami własności, gdyż nie zamierzam porzucać mej wolności dla takiego drobiazgu. Zgodnie z zasadą, którą wyznaję, nie chcę korzystać z czegoś, czym nie mogę podzielić się z innymi".

Tego rodzaju odpowiedzi podkreślają quasireligijną naturę jego przesłania. Jak żyd przestrzegający koszeru lub mormon niepijący alkoholu, Stallman swoją decyzję niekorzystania z oprogramowania z prywatnym prawem własności ubiera w szaty tradycji i osobistych wierzeń. Jako niosący nową programistyczną ewangelię unika narzucania swych wierzeń sił a, jednak słuchacze rzadko wychodzą z jego wykładów nie przekonani, gdzie leży prawda.

Aby dotrzeć ze swym przesłaniem do słuchaczy, uzupełnił wykład rytualnym przerywnikiem. Z plastikowej torby wyciągnął czarną togę i nałożył ją na ramiona, z drugiej torby wyjął malowany żółtą odblaskową farbą dysk komputerowy i umieścił go sobie na głowie, co widownia przyjął a ze śmiechem. "Jestem święty Ignacy z Kościoła Emacs - powiedział, wznosząc prawą rękę w geście sztydlerczego błogosławieństwa. - Błogosławie twoj komputer, moje dziecko".

[Rysunek0801]

Stallman jako święty Ignacy. Zdjęcia Wouter van Oortmerssen

Słuchacze wybuchnęli niepoohamowanym śmiechem. Widownia klaskała, dysk na głowie Stallmana chwytal światło lamp świecących nad głową i poblyskiwał, doskonale imitując aureolę. W jednej chwili przeistoczył się z niezręcznego haole21 w postać z rosyjskiej świętej ikony.

"Emacs został napisany jako zwykły edytor tekstowy - mówił Stallman, tłumacząc swój strój. - Z czasem stał się sposobem na życie dla wielu i religią dla niektórych. Nazywamy tę religię Kościołem Emacs".

Ten moment autoparodii rozweselił zebranych, trafiając również do tych wielu, którzy wcześniej stallmanowską formę programistycznego ascetyzmu zdawali się uważać za

maskowany przebraniem fanatyzm religijny. Był oto także gł ősne "zrzucenie drugiego buta"²². Nałożenie togi i aureoli był oprzyzwoleniem: "Dobrze, śmieJCie się ze mnie. Wiem, że wyglądam dziwacznie".

A co się tyczy samej osoby świętego Ignucego, Stallman twierdzi, że pojawił się on w roku 1996, długo po powstaniu Emacsa i równie długo przed pojawieniem się terminu "oprogramowanie open source". Jego pojawienie się zostało przyspieszone dążeniem do objęcia przywództwa hackerskiej społeczności. Mówiąc w ten sposób "śmieJCie się ze mnie", chciał przekonać wszystkich, że on, uparty Stallman, nie jest fanatykiem, choć wielu stara się przedstawić go w ten sposób. Dodaje, że później inni starali się nadać temu zbyt wielkie znaczenie, by podkreślić jego rolę programistycznego ideologa, jak to zrobił Eric Raymond w wywiadzie z roku 1999 umieszczonym w serwisie internetowym linux.com:

Gdy mówię, że RMS skaluje swoje poczynania, nie lekceważę go i nie oskarżam o nieszczer ősć. Stwierdzam jedynie, że jak każdy dobry "przekonywacz" stosuje teatralne gesty. Czasem robi to świadomie dla zwiększenia efektu - czyż nie widzieliście go, jak przebrany za świętego Ignucego z dyskową aureolą na głowie błogosławi oprogramowanie? Jednakże często czyni to nieświadomie; umie odpowiednio pobudzić słuchaczy, aby trzymać ich uwagę w napięciu, zwykle unikając szokowania²³. Stallman analizę Raymonda kwituje stwierdzeniem: "To mój sposób śmiania się z samego siebie. Jeżeli inni widzą w tym coś więcej, jest to ich problem, a nie mój".

Przyznaje się do szmiry. "Śmiesz was to? - zapytał w pewnym momencie. - Uwielbiam być w centrum zainteresowania". Jak mówi, aby sobie to ułatwić, zapisał się kiedyś do Toastmasters, organizacji pomagającej swoim członkom udoskonalać umiejętność publicznego występowania, i teraz wszystkim to poleca. Osiągnął umiejętności sceniczne, których mogą mu pozazdrościć profesjonalni aktorzy i wyczuwa związek swych wystąpień z wodewilami ostatnich lat. W kilka dni po

przemówieniu Stallmana w Maui High Performance Computing Center, robiąc aluzję do wystąpienia na Linux World Expo 1999, spytał emgo, czy ma kompleks Groucho Marxa, tzn. niechęć do należenia do wszelkich klubów, które chętnie widział yby go jako swego członka. Odpowiedział natychmiast: "Nie, ale podziwiam Groucho Marxa i sędzę, że pod wieloma względami jestem inspirowany przez niego, a także przez Harpo²⁴."

Wpł ywGroucho Marxa jest widoczny w jego zamiłowaniu do gier słownych, choć z drugiej strony jest to typowa cecha większości hakerów. Wpł ywGroucha daje się zauważyć przede wszystkim w sposobie wypowiadania kalamburów. Mówi je z kamienną twarzą, jakby mimochodem, bez zdradzania emocji nawet przez uniesienie brwi lub wykrzywienie ust, co sprawia wrażenie, że on bardziej śmieje się z widowni, niż ona z niego. Ale wrażenie to szybko znika, gdy patrzymy na członków zespołu Maui High Performance Computer Center, śmiejących się z parodii świętego Ignucego. Bez żadnego szczególnego wysiłku potrafił skłonić do śmiechu salę peł ną inżynierów. "Bycie świętym Kościoła Emacs nie wymaga celibatu, ale zmusza do życia w moralnej czystości - mówił do mauikańskiego audytorium. - Musicie egzorcyzmami wygnać diabelski system operacyjny z prawem wł aśności i zainstalować na waszych komputerach cał kowicie wolny i święty system GNU, a na nim instalować jedynie wolne oprogramowanie. Jeżeli złożycie takie śluby i będziecie żyć, nie ł ańiać ich, zostaniecie świętymi Kościoła Emacs i aureola pojawi się wokół waszych głów.

Skecz o świętym Ignucym zakończył dowcipem z podtekstem. W większości systemów uniksowych i spokrewnionych z Uniksem gł ównym konkurentem Emacsa jest vi, edytor tekstowy stworzony przez byłego studenta UC Berkeley²⁵, a obecnego szefa działu badawczego Sun Microsystems, Billa Joya. Przed zdjęciem aureoli Stallman pokpiwał sobie z programu rywala. Mówił : "Ludzie czasami pytają mnie, czy grzechem wobec

Kościół Emacs jest używanie vi. Odpowiadam, że korzystanie z wolnej wersji vi to nie grzech, lecz pokuta. Szczęśliwego hakowania".

Po krótkiej sesji zadawania pytań słuchacze zebrali się wokół Stallmana. Kilku poprosiło o autografy. "Podpiszę to - powiedział, trzymając podany przez jakąś kobietę wydruk Powszechnej Licencji Publicznej GNU26 - ale tylko wtedy, gdy przyrzekniesz mi, że będziesz używać prawidłowej nazwy GNU/Linux zamiast Linux i wszystkim swoim przyjaciołom nakażesz to samo".

Ten komentarz jedynie potwierdza moje prywatne obserwacje. W przeciwieństwie do innych występujących na scenie oraz wielu polityków, Stallman nie ma "trybu prywatnego". Obok świętego Ignacego istnieje ideolog, taki sam na scenie, jak i poza nią. Tego samego wieczoru podczas rozmowy przy obiedzie, gdy jakiś programista wspomniał o swoim pozytywnym stosunku do oprogramowania open source, Stallman poprawił go: "Masz na myśli wolne oprogramowanie. Tak należy o nim mówić."

Zadającym pytania po wystąpieniu Stallman przyznał się do zapędów pedagogicznych. "Wielu ludzi powiada: 'Najpierw zaprosz do wstąpienia do społeczności, a potem nauczaj o wolności'. Może to być rozsądna strategia, ale w praktyce wszyscy zapraszają do udziału w społeczności, ale gdy chętni już się pojawiają, mało kto zajmuje się nauczaniem o wolności". Przypomina to nieco miasto z trzeciego świata. Ludzie tam ciągną z nadzieją wzbogacenia się lub przynajmniej uczestniczenia w otwartej, żywej kulturze, ale ci, którzy dzierżą władzę, wciąż stosują różne triki i strategie, np. patenty chroniące oprogramowanie, aby trzymać masy z dala. "Wiele milionów ludzi buduje miasta slamsów, ale nikt nie dba o postąpienie o krok naprzód, by ich z tych ruder wyciągnąć. Jeżeli pozytywnie oceniacie strategię wolnego oprogramowania, proszę, wykonajcie następny krok. Krok pierwszy wykonuje wielu. Potrzeba nam tych, którzy postąpią dalej."

Wykonanie drugiego kroku, czyli niesienie do domów idei wolności, a nie tylko jej akceptacja, jest kamieniem węgielnym ruchu wolnego oprogramowania. Ci, którzy mają nadzieję na zreformowanie od środka przemysłu programistycznego opartego na zasadach własności, grzeszą głupotą. "Zmienianie systemu od środka jest ryzykowne - twierdził Stallman - i jeżeli nie działacie z pozycji Gorbaczowa, zostaniecie unicestwieni". Chętni do zadawania pytań podnosili ręce. Stallman zwrócił uwagę na kogoś w stroju klubu golfowego, kto pytał: "Więc jak bez patentów można walczyć ze szpiegostwem przemysłowym?" "Te dwie sprawy nie mają ze sobą nic wspólnego, naprawdę" - odpowiedział Stallman.

"A jeżeli ktoś chce wykraść z innej firmy część oprogramowania?"

Stallman wyglądał jak ktoś rażony piorunem. "Chwileczkę - powiedział. - Wykraść? Przepraszam, ale ta wypowiedź zawiera tak wielki ładunek przesad, że mogę jedynie powiedzieć, że ich nie podzielam. Firmy, które tworzą "niewolne" oprogramowanie i inne rzeczy, skrywają wiele sekretów handlowych i nie wygląda na to, by tutaj mógł coś się zmienić. W latach osiemdziesiątych znaczna część programistów nawet nie miała pojęcia, czym jest patent chroniący oprogramowanie, więc nie przykładała do tego żadnej wagi. Ludzie publikowali interesujące idee i jeżeli nie należeli do ruchu wolnego oprogramowania, zachowywali w sekrecie niewielkie szczegóły. Teraz opatentowują te idee, a drobne szczegóły nadal zachowują w tajemnicy. Tak czy inaczej, patenty w opisywanej dziedzinie nic nie zmieniły."

"Więc skoro nie wpłynęła na publikowanie..." - głos kolejnej osoby, która wtrącił się do dyskusji, zamarł niemal na początku.

"Ależ wpłynęła - odpowiedział Stallman. - Teraz publikacje informują was, że taki a taki pomysł jest odcięty od społeczeństwa na 20 lat. I co w tym, do diabła, widzicie dobrego? Poza tym opisują wszystko w możliwie najbardziej

niejasny sposób, aby rozciągnąć patent możliwie najszerzej i w rezultacie czytanie dokumentacji traci sens, bo i tak nic nie można zrozumieć. Czytanie patentów przynosi jedynie złe informacje o zdarzeniach, z którymi walka jest daremna". Na sali zapadła cisza. Wystąpienie rozpoczęło się o 15:15, a był a już niemal 17:00 i większość słuchaczy wierciła się na siedzeniach, niecierpliwie oczekując, by wreszcie zerwać się i uciec na weekend. Wyczuwając zmęczenie, Stallman rozejrzał się po sali i szybko zakończył. "Wygląda na to, że skończyliśmy - powiedział, po czym na wzór prowadzącego aukcję, zaczął odliczać - "Koniec po raz pierwszy, koniec po raz drugi, koniec po raz trzeci. Wykład zamknięty". W ten sposób dał słuchaczom możliwość zadania ewentualnych ostatnich pytań. Skoro już żadne ręce nie podniosły się do góry, zakończył tradycyjnym: "Szczęśliwego hakowania".

Przypisy

- 1 Mauitańskie centrum komputerowe o wysokiej wydajności - przyp. tł um.
- 2 Pola golfowe Silversword - przyp. tłum.
- 3 Argonne National Laboratory (University of Chicago) - <http://www.anl.gov> - przyp. tł um.
- 4 Słynne laboratorium, w którym w czasach II wojny światowej międzynarodowy zespół naukowców pod kierownictwem Roberta Oppenheimera zrealizował Projekt Manhattan, czyli budowę pierwszej bomby atomowej - <http://www.lanl.gov/worldview/> - przyp. tł um.
- 5 Palm Pilot bywa nazywany komputerem naręcznym, ale niektórzy tę nazwę zastrzegają dla komputerków nieco większych, używanych ostatnio przez różnego rodzaju inkasentów i mierniczych. Dlatego zdecydowałem się podać nazwę angielską, a ciekawych i niezorientowanych odsyłam na stronę <http://www.palm.com>, gdzie mogą obejrzeć Palm Piloty

w pełnej krasie

- przyp. tłum.

6 Pochodzący z głównego ładu (kraju) - przyp. tłum.

7 Mauitańska grupa użytkowników FreeBSD - przyp. tłum.

8 Popularny zespół muzyczny. Informacje o nim można znaleźć pod adresem <http://grateful.dead.net/> - przyp. tłum.

9 Richard Matthew Stallman - przyp. tłum.

10 General Public Licence - patrz przypis 7. w rozdziale 2. - przyp. tłum.

11 Właściwie Magna Charta Liberatum (łac.) - Wielka Karta Swobód była aktem wydanym w Anglii przez Jana Bez Ziemi w 1215 r. Przyznawał a przywileje możnowładztwu (kosztem władzy królewskiej) i pewne prawa stanom niższym; w Anglii jest uważana za fundament wolności obywatelskich - przyp. tłum.

12 Patrz: <http://www.zdnet.com/> - przyp. tłum.

13 Evan Leibovitch, Who's Afraid of Big Bad Wolves, ZDNet Tech Update (15 grudnia 2000).

14 Dla celów narracyjnych powinienem zagłębić się w opis stallmanowskiej definicji wolnego oprogramowania, ale zawahałem się przed tym i cytuję ją w pełnej postaci według serwisu WWW projektu GNU. Zostały tam wyliczone cztery zasadnicze cechy wolności oprogramowania:

The freedom to run a program, for any purpose (freedom 0).

The freedom to study how a program works, and adapt it to your needs (freedom 1).

The freedom to redistribute copies of a program so you can help your neighbor (freedom 2).

The freedom to improve the program, and release your improvements to the public, so that the whole community benefits (freedom 3).

Wiecej informacji można znaleźć na stronie "The Free Software Definition" pod adresem <http://www.gnu.org/philosophy/free-sw.html>.

15 Na polskiej stronie internetowej projektu GNU

<http://www.gnu.org/philosophy/free-sw.pl.html> zostało to

przetł umaczone następująco:

... mówimy o czterech rodzajach wolności użytkowników programu:

wolność uruchamiania programu w dowolnym celu (wolność 0);

wolność analizowania, jak program działa, i dostosowywania go do swoich potrzeb (wolność 1). Warunkiem koniecznym jest tu dostęp do kodu źródłowego;

wolność rozpowszechniania kopii, byście mogli pomóc sąsiadom (wolność 2);

wolność udoskonalania programu i publicznego rozpowszechniania własnych ulepszeń, dzięki czemu może z nich skorzystać cała społeczność (wolność 3). Warunkiem koniecznym jest tu dostęp do kodu źródłowego.

- przyp. tłum.

16 Dokładniej Clean-Room Reverse-Engineering, co mniej więcej można przełożyć na polski jako "inżynieria wsteczna wykonywana w czystym pokoju" lub dokładniej, ale mniej zgrabnie jako "czystopokojowa inżynieria wsteczna". Mówimy o inżynierii wstecznej typu "clean room", jeżeli twórcy oprogramowania naśladowującego starają się unikać dokładnego studiowania oryginalnego programu i jego publikowanych specyfikacji. Nie wykonują również dokładnych testów oryginalnego programu. Patrz np.:

<http://www.ardi.com/reveng.php> - przyp. tłum.

17 Tutaj Stallman zaczyna toczyć spór ze zwolennikami ruchu oprogramowania open source. Należy zauważyć, że na rzecz udostępniania użytkownikom kodów źródłowych działają w świecie dwa niezależne ruchy społeczne: Free Software i Open Source, co po polsku przyjęto nazywać odpowiednio "wolnym oprogramowaniem" i "oprogramowaniem open source" (inaczej oprogramowaniem otwartych kodów źródłowych, oprogramowaniem wolnodostępnych źródeł - istnieje kilka tłumaczeń, ale najczęściej mówi się jednak o oprogramowaniu open source). Pierwszym kieruje Free Software Foundation

(Fundacja Wolnego Oprogramowania), o której już w książce był a mowa, a drugim Open Source Initiative (co dokładnie należałoby przełożyć jako Inicjatywa Otwartych Źródeł). Istnieje wyraźna rozbieżność między filozofiami "free software" i "open source". "Free software" kładzie główny nacisk na stronę moralną i etyczną dostępności oprogramowania, natomiast "open source" podkreśla znaczenie technicznej doskonałości kodu. Sporo informacji na ten temat znajduje się w rozdziale 11. - przyp. tłum.

18 Patrz <http://www.opensource.org/> - przyp. tłum.

19 Eric Raymond, Shut Up and Show Them the Code, esej internetowy, (28 czerwca 1999).

20 Ibidem.

21 Tak na Hawajach nazywają białych przybyszów - przyp. tłum.

22 The other shoe dropping - nie znam polskiego odpowiednika tego powiedzenia, więc tłumaczę dosłownie, przytaczam w oryginale i opatruję wyjaśniającym przypisem. Zwrot ten oznacza zakończenie jakiegoś działania, będące jednocześnie poniesieniem konsekwencji wcześniejszych poczynań. Źródłem powiedzenia jest stary dowcip amerykański, w którym znużonego wędrowca, wynajmującego pokój w przydrożnym motelu, recepcjonista ostrzegł, by nie hałasował i nie budził innych podróżnych. Zmęczony jego mową, kładąc się spać, przez nieuwagę lub zapomnienie z hukiem zrzucił z nogi jeden but. Przerażony tym, drugi but niezwykle ostrożnie i cicho ułożył na podłodze. Gdy przysnął, rozległ o się pukanie. Stojąca za drzwiami grupa gości hotelowych prosiła, by wreszcie zrzucił drugi but, bo czekając na to, nie mogą usnąć - przyp. tłum.

23 Guest Interview: Eric S. Raymond, Linux.com (18 maja 1999), <http://www.linux.com/interviews/19990518/8/>

24 Marx Bracia - Chico (właściwie Leonard, 1891 - 1961), Harpo (Adolph, zwany Arthurem, 1893 - 1964), Groucho (Julius Henry, 1895 - 1977). Początkowo również Gummo

(Milton, 1893 - 1977 i Zeppo (Herbert, 1901 - 1979).

Występowali z matką w wodewilach jako tancerze i piosenkarze. Odnieśli sukces na Broadwayu i - od 1929 - w Hollywood.

Trzej pierwsi bracia utworzyli zespół utalentowanych komików-muzyków, których purnonsensowy humor przenosił widza w świat surrealistycznej groteski. W każdym filmie stosowali przerywniki muzyczne, demonstrując: Chico - wirtuozerskie wariacje fortepianowe, a Harpo grę na harfie.

Wystąpili w filmach: Małpi interes (1931), Końskie pióra (1932), Kacza zupa (1933), Noc w operze (1935), Dzień na wyścigach (1937), Noc w Casablance (1946), Niewiarygodna kradzież diamentów (1959) i in. Groucho zagrał samodzielnie w filmach: Mr Music (1950), Skidoo (1968) i in. Pracował również w telewizji. W 1973 otrzymał Oscara w uznaniu zasług własných i całego zespołu - przyp. tłum. na podstawie encyklopedii internetowej <http://wiem.onet.pl>

25 Berkeley University of California - Uniwersytet Kalifornijski w Berkeley. Berkeley należy do zespołu miejskiego San Francisco, patrz <http://www.berkeley.edu/> - przyp. tłum.

26 Patrz przypis 6. w rozdziale 2. - przyp. tłum.

9. POWSZECHNA LICENCJA PUBLICZNA GNU1

Do wiosny 1985 r. Richard Stallman uporał się ze stworzeniem pierwszego kamienia milowego projektu GNU, czyli opartej na Lispie wolnej wersji Emacsa. Aby tego dokonać, musiał sprostać dwóm wyzwaniom. Po pierwsze - uniezależnić Emacs od platformy systemowej, a po drugie - w tym duchu odtworzyć społeczność Emacs.

Dysputy z UniPressem wyciągnęły na światło oienne wady kontraktu wewnętrznego społeczności Emacs. Tam, gdzie Emacs rozwijał się pod bezpośrednim nadzorem Stallmana, zasady umowy społecznej były dotrzymane, ale w działach rozwiniętych bez jego kierownictwa i ogólnego nadzoru, np. dotyczących wersji Uniksa sprzed roku 1984, poszczególne

osoby i firmy miały swobodę ustanawiania własnych zasad. Napięcie między swobodą modyfikacji i swobodą narzucana przywilejów autorskich zrodziło się jeszcze przed GOSMACS-em. Copyright Act (Akt praw autorskich) z roku 1976 rozciągał na oprogramowanie prawa autorskie regulujące zasady kopiowania. Zgodnie z ustępem 102(b) tegoż aktu, osoby i firmy uzyskały możliwość chronienia prawem autorskim (copyright) samych programów, ale nie "procesów i metod w ten program wbudowanych"². Inaczej mówiąc, programy zostały potraktowane tak samo jak opowiadania i piosenki. Inni programiści mogli czerpać inspirację z istniejących programów, ale aby je skopiować, musieli - i to nie żart - postarać się wcześniej o zgodę autora oryginału. Choć nowe prawo gwarantowało ochronę prawną również programom bez noty copyright, programiści szybko przystąpili do zaznaczenia swych praw przez dodawanie tej noty.

Stallman od początku patrzył na to z przerażeniem. Rzadko się zdarza, by jakiś program nie zapożyczał części kodu od programów już istniejących, a tu jedno pociągnięcie prezydenckiego pióra i jedna uchwała Kongresu dały programistom i firmom programistycznym przewagę nad społecznie budowanymi programami. Ponadto był to silny zastrzyk formalizmu zaaplikowany światu, który dotychczas pozostawał wolny od biurokracji. Nawet jeżeli hakerzy byliby w stanie udowodnić, że kod jakiegoś przeklętego programu korzysta z działań realizowanych przez lata, taka batalia z posiadaczami praw własności przekraczała ich możliwości finansowe. Dawne dyskusje hakera z hakerem zostały teraz zastąpione przez negocjacje prawnika z prawnikiem. Z takiego systemu prawnego ciągnęły korzyści przede wszystkim firmy, a hakerzy nie mieli szans.

Ci, którzy zaproponowali ochronę praw autorskich oprogramowania, wysuwali kontrargumenty: bez ochrony praw autorskich rezultaty pracy mogły się przedostawać do domeny publicznej. Ponadto nota copyright była gwarancją jakości.

Programiści i firmy, umieszczając swe imię lub nazwę w nocie autorskiej, pracowali na swą reputację. Ostatecznie był to rodzaj kontraktu, jak również stwierdzenie prawa własności. Używając noty copyright jako elastycznej postaci licencji, autor mógł rzec się części swych praw w zamian za określone zachowanie użytkownika. Autor mógł też przykłać do zezwolić na nieautoryzowane kopiowanie programu pod warunkiem, że nie będzie on wykorzystywany w celach komercyjnych.

Ten ostatni argument nieco łagodził sprzeciw Stallmana wobec stosowania noty copyright. Patrząc wstecz na lata prowadzące do rozpoczęcia projektu GNU, przyznaje on, że zaczął rozumieć korzyści płynące ze stosowania noty copyright gdzieś na etapie wersji Emacs 15.0, ostatniej ważnej poprzedzającej powstanie GNU Emacs. Wspomina: "Widziałem e-maile z uproszczonymi notami copyrightu: 'Kopiowanie bez zmian dozwolone' i to było inspirujące".

Dla wersji Emacs 15.0 stworzył notę copyright dającą użytkownikom prawo sporządzania i upowszechniania kopii. Dał im również prawo tworzenia wersji zmodyfikowanych, ale bez możliwości przypisywania sobie wyłącznego autorstwa, jak to miało miejsce w przypadku GOSMACS.

Licencja Emacs 15.0, choć pomocna przy kodyfikowaniu umowy wzajemnej społeczności Emacs, okazała się zbyt nieformalna, aby się nią posłużyć w projekcie GNU. Zaraz po rozpoczęciu pracy nad wersją GNU Emacs Stallman rozpoczął konsultację z innymi członkami Fundacji Wolnego Oprogramowania nad dopracowaniem języka licencji. Szukał także rady u prawników, którzy pomagali mu stworzyć Fundację.

Mark Fisher, adwokat bostoński specjalizujący się w prawie własności intelektualnej, przypomina sobie z tego okresu dyskusje ze Stallmanem na temat postaci licencji. "Richard miał bardzo sprecyzowaną wizję, jak ta licencja ma wyglądać - wspomina. - Miała opierać się na dwóch podstawowych zasadach. Po pierwsze, licencja miała czynić oprogramowanie

maksymalnie dostępnym (otwartym). Po drugie, miał zachęcać innych do stosowania tych samych praktyk licencyjnych." Zachęcanie innych do stosowania tego samego sposobu licencjonowania oprogramowania oznaczało ograniczenie powstawania nowych wersji Emacs'a z zastrzeżonymi prawami własności. Aby ukrócić ten proceder, Stallman z grupą kolegów programistów, zwolenników wolnego oprogramowania ogłosili, że każdy użytkownik może do woli modyfikować GNU Emacs'a dopóty, dopóki będzie publikował zmiany. Dodatkowo każda nowa, pochodna wersja musiała być opatrzona tą samą licencją GNU.

Rewolucyjna natura tego ostatniego zastrzeżenia ujawnił się dopiero po pewnym czasie. Fischer wspomina, że wówczas patrzył na licencję GNU Emacs jak na zwykły dość prosty kontrakt. Dla towaru został określona cena. Jediną różnicą był to, że zamiast pobierania opłaty pieniężnej Stallman zażądał od użytkowników udostępniania innym wszelkich wprowadzanych zmian. To, jak pamięta Fischer, odróżniało ten kontrakt od innych.

"W tym czasie samo proszenie innych o zaakceptowanie proponowanej ceny był orzeczą unikalną" - twierdzi.

Licencja GNU Emacs ostatecznie pojawiła się w roku 1985 wraz z samym programem GNU Emacs. Po ukazaniu się programu Stallman zwrócił się do hakerskiej społeczności o pomoc w ulepszeniu licencji. Rychło na apel odpowiedział John Gilmore, haker pracujący dla Sun Microsystems, który stał się potem aktywistą ruchu wolnego oprogramowania. Jako swój wkład Gilmore wniósł wersję Emacs'a przystosowaną do SunOS, wewnątrz formowej wersji Uniksa. Podczas adaptacji wykonał dwie poprawki zgodnie z wymaganiami licencji. Uznał tę licencję nie za obciążenie, lecz za zwieńczenie hakerskiego etosu. "Wszystkie dotychczasowe licencje były bardzo nieformalne" - wspomina.

Jako przykład braku formalizmu Gilmore cytuje notę copyrightu dla trn, programu narzędziowego Uniksa. Program

ten, napisany przez Larry' eg Walla, późniejszego twórcę języka Perl, ułatwiał programistom uniksowym wprowadzanie do dużych kodów źródłowych poprawek, tzw. "patchów".

Objaśniając przydatność tego programu, Wall w tekstowym pliku README dopisał :

Copyright (c) 1985, Larry Wall

Możecie trn kopiować w całości albo po kawałku, dopóki nie robicie na tym forsy lub nie udajecie, że to wyście go napisali³. Takie stwierdzenia, będące doskonałym odbiciem hakerskiej etyki, jednocześnie ilustrowały trudności przetłumaczenia jej swobodnej postaci na sztywny, formalny język legalnego copyrightu. Pisząc licencję dla GNU Emacs, Stallman zrobił coś więcej niż proste zablokowanie możliwości wymykania się zmodyfikowanych wersji na rynek oprogramowania z prywatnymi prawami własności. Wyraził zasady hakerskiej etyki w języku zrozumiałym dla obu stron: hakerów i prawników.

Nie upłynęło wiele czasu, nim inni hakerzy zaczęli rozmyślać nad "przyczepieniem" licencji GNU Emacs do swoich własnych programów. W odpowiedzi na wezwanie zgłoszone w usenetowej dyskusji Gilmore w listopadzie 1986 roku wysłał do Stallmana e-mail sugerujący wprowadzenie dwóch modyfikacji:

W nazwie licencji mógłbyś chyba "EMACS" zastąpić przez "SOFTWARE" lub coś podobnego. Mam nadzieję, że wkrótce Emacs przestanie być największą częścią systemu GNU i licencja będzie obejmować wszystko⁴.

Nie tylko Gilmore proponował większą ogólnikowość licencji. Do końca roku 1986 Stallman osiągnął kolejny kamień milowy projektu GNU - debugger kodu źródłowego - i szukał sposobu zmodyfikowania licencji Emacs tak, aby objęła oba programy. Zdecydował się na usunięcie wszystkich sformułowań odnoszących się bezpośrednio do Emacsa i stworzenie ogólnej licencji, jednego ogólnego "parasola", który przykryje całe oprogramowanie projektu GNU. W ten sposób narodził się Powszechna Licencja Publiczna GNU (GNU General Public

License, czyli w skrócie GPL).

Do oznaczania licencji przyjął powszechnie stosowaną konwencję numerowania wersji oprogramowania, zgodnie z którą pełne wersje oznacza się liczbami całkowitymi, zaś prototypowe za pomocą ułamków dziesiętnych. GPL w wersji 1.0 opublikował w roku 1989 (prace nad samym projektem trwał od roku 1985), czyli w rok po wypuszczeniu GNU Debuggera (drugiego wejścia w świat programowania uniksowego). Licencja zawierała apreambulę wyrażającą intencje polityczne:

Celem Powszechnej Licencji Publicznej GNU (GNU General Public License, GPL) jest zagwarantowanie użytkownikowi swobody udostępniania i sprzedawania kopii wolnego oprogramowania, swobody otrzymywania kodu źródłowego, jeżeli jest potrzebny, i swobody zmieniania wolnego oprogramowania, a więc danie pewności, iż oprogramowanie jest wolno dostępne dla wszystkich użytkowników.

W celu ochrony praw użytkownika jesteśmy zmuszeni wprowadzać ograniczenia zabraniające komukolwiek kwestionowanie jego praw albo sugerowanie rezygnacji z tych praw. Ograniczenia te sprowadzają się do pewnych obowiązków w przypadku rozpowszechniania przez Was kopii naszego oprogramowania bądź dokonywania w nim zmian⁵ i ⁶.

Przygotowując GPL, Stallman był zmuszony wprowadzić zmiany przystosowujące licencję do nieformalnych zasad obowiązujących wewnątrz społeczności Emacs. Poprzednie żądanie publikowania wszystkich wprowadzanych zmian musiał zamienić na żądanie publikowania jedynie zmian wprowadzonych w programach pochodnych, upowszechnianych w taki sposób, jak on to czynił. Mówiąc inaczej, programiści, którzy modyfikowali Emacs jedynie dla własnych potrzeb, zostali zwolnieni z przesyłania do niego kopii plików źródłowych. To obniżenie ceny wolnego

oprogramowania był o jednym z niewielu zaaprobowanych przez niego odstępstw od doktryny. Teraz użytkownicy mogli wprowadzać zmiany bez obawy, że Stallman zagląda im przez ramię, jeżeli tylko nie odcinali reszty hakerskiej społeczności od późniejszych zmian danego programu.

Patrzac wstecz, Stallman przyznaje, że kompromisowe sformułowanie wprowadził do GPL, gdyż nie podobało mu się pełnienie roli Wielkiego Brata wobec społeczności Emacs. Lubił zaglądać w systemy innych hakerów, ale obawiał się, że w przyszłości ktoś inny, nadzorując rozwój kodu, może taką uprzywilejowaną pozycję wykorzystać w złym celu. To właśnie skłoniło go do złagodzenia warunków GPL.

"Nie jest dobre zmuszanie ludzi do publikowania wszystkich zmian - mówi Stallman. - Nie jest słuszne żądanie, by wszystkie informacje przesyłać alido jednego uprzywilejowanego użytkownika. Taki rodzaj centralizacji i uprzywilejowania jednej osoby nie jest zgodny z zasadami społeczeństwa, w którym wszyscy mają równe prawa".

Powszechna Licencja Publiczna (GPL) może być uznana za największe hakerskie osiągnięcie Stallmana. Stworzył a ona system własności społecznej w ramach powszechnie obowiązującego systemu prawa autorskiego (copyright). A co jeszcze ważniejsze, ukazała intelektualne podobieństwo tekstu zapisu prawnego i kodu programowego. Preambuła GPL zawiera głębokie przesłanie, że zamiast spoglądać z podejrzliwością na prawo autorskie (copyright), hakerzy mogą je uważać za kolejny system aż się proszący o "zhakowanie".

"GPL powstawał ajak fragment wolnego oprogramowania, tworzony przez dyskusję dużej społeczności, która omawia ogólną konstrukcję, zgodność lub niezgodność z własnymi wymaganiami, domaga się złagodzenia różnych elementów i dojścia do kompromisu zadawalającego możliwie dużą grupę - mówi Jerry Cohen, inny prawnik, który pomagał Stallmanowi przy tworzeniu licencji. - Proces ten przebiegał bardzo sprawnie i choć pierwsze opublikowane wersje zostały przyjęte ze

sceptycyzmem, a nawet wrogością, kolejne były coraz szerzej akceptowane."

W roku 1986 w wywiadzie dla magazynu "Byte" Stallman optymistycznie przedstawił wyniki pracy nad GPL. Oprócz proklamacji wartości hakerskich czytelnicy powinni odebrać tę licencję jako "rodzaj intelektualnego pojedynku jujitsu z systemem prawnym stworzonym przez tych, którzy oprogramowanie chcą zamienić w swój tajny kapitał"7. Lata później Stallman opisuje powstanie GPL w słowach o wiele mniej wojowniczych: "Rozmyślałem o materii, która miała sens etyczny, polityczny i prawny - mówi. - Musiałem zastanowić się, które elementy systemu prawnego należy w licencji pozostawić. Wszak przystępowałem do tworzenia podstaw prawnych nowej społeczności, a nie byłem rządem i nie mogłem zmieniać istniejącego prawa. Zadanie musiałem wykonać, tworząc budowlę na szczycie istniejącego systemu prawnego, który nie był stworzony z myślą o realizacji celów, jakie mi przyświecały."

W czasie, gdy on ważył dotyczące wolnego oprogramowania racje etyczne, polityczne i prawne, kalifornijski haker i członek klubu entuzjastów science-fiction, Don Hopkins, odesłał mu pożyczoną wcześniej instrukcję mikroprocesora 68000. W dowód wdzięczności udekorował tył koperty nalepkami otrzymanymi na lokalnym zjeździe miłośników science-fiction. Jedna z nalepek przyciągnęła wzrok Stallmana. Widniał na niej napis "Copyleft (L), All Rights Reversed8". Wydając pierwszą wersję GPL, Stallman zapożyczył termin z tekstu nalepki, nadając licencji wolnego oprogramowania nieformalną nazwę "Copyleft"9. Z czasem nazwa nieformalna i jej skrót "odwrócone C", stał się dla Fundacji Wolnego Oprogramowania symbolem Powszechnej Licencji Publicznej (GPL).

Niemiecki socjolog Max Weber stwierdził kiedyś, że wszystkie wielkie religie wyrosły z "rutynizacji" i instytucjonalizacji charyzmy. Według niego, każda religia, która zdobyła sobie trwałe miejsce w historii, osiągnęła to przez przekształcenie

charyzmy i posłania swego przywódcy-proroka w społeczny, polityczny i etyczny aparat głoszący własne prawdy w formie możliwej do przyjęcia przez różne kultury w różnych czasach. Choć GNU GPL per se nie jest przesłaniem religijnym, na pewno można uważać ją za interesujący przykład procesu instytucjonalizacji następującej w nowoczesnym, zdecentralizowanym świecie twórców oprogramowania. Wkrótce po pojawieniu się licencji od razu zaakceptowała ją większość programistów i firm, nawet niepoczuwających się do jakiegokolwiek wspólnoty z ruchem Stallmana. Niektórzy uznali GPL za podstawę ochrony własnego oprogramowania. Nawet ci, którzy odrzucili GPL jako zbyt ograniczającą, przyznali, że pozostali pod jej wpływem.

Do drugiej grupy należał haker Keith Bostic, w chwili opublikowania GPL 1.0 pracownik University of California. Dział, w którym pracował - Computer Systems Research Group (CSRG)¹⁰ - zajmował się rozwojem oprogramowania uniksowego od roku 1970 i był odpowiedzialny za wiele kluczowych elementów systemu, włączając w to protokół sieciowy TCP/IP, podstawę nowoczesnej komunikacji internetowej. Pod koniec lat osiemdziesiątych AT&T, właściciel nazwy firmowej Unix, zaczął się interesować komercjalizacją tego systemu i począł spoglądać na Berkeley Software Distribution (BSD), akademicką wersję Uniksa, rozwiniętą przez Bostica i jego kolegów, jak na kluczowe źródło komercyjnych technologii.

Choć na podstawie licencji kod źródłowy BSD był dostępny dla naukowców i komercyjnych programistów, te usiłowania komercjalizacji zaczęły stawać się problemem. Prawa własności kodu Berkeley i kodu komercyjnego AT&T były wymieszane. W rezultacie dystrybucje Berkeley były dostępne jedynie dla instytucji, które miały licencję kodu źródłowego AT&T. Gdy AT&T podniosło ceny licencji, układ, który z początku wydawał się nieszkodliwy, stopniowo stawał się coraz cięższy do zniesienia.

Przyjęty do pracy w roku 1986, Bostic przyjął na siebie zadanie przystosowania BSD do działania na komputerze PDP-11 firmy Digital Equipment Corporation. Wówczas, jak mówi, dość blisko współdziałał ze Stallmanem podczas jego okazjonalnych wizyt na zachodnim wybrzeżu. "Pamiętam, że Stallman siedział przy terminalu CSRG, gdy toczyliśmy żywą wymianę zdań na temat praw autorskich - wspomina. - Potem poszliśmy na obiad i dalej dyskutowali na ten sam temat".

Jego argumenty ostatecznie zostały przyjęte, choć niedokładnie w takiej formie, jak sobie życzył. W czerwcu 1989 Berkeley oddzieliło swój kod sieciowy (BSD Networking) od reszty systemu operacyjnego, do którego prawa własności miał AT&T, i zaczął ogłaszać go z niezależną licencją University of California. Warunki kontraktu były bardzo liberalne. Licencjobiorca był zobowiązany jedynie do powoływania się na uczelnię w ogłoszeniach reklamujących programy pochodne. W przeciwieństwie do GPL licencja ta zezwalała na tworzenie programów pochodnych z zastrzeżonymi prawami własności. Powszechnemu uznaniu licencji BSD przeszkadzał fakt, że wersja BSD Networking nie była pełnym systemem operacyjnym. Można było studiować jego kod, ale jego uruchomienie wymagało połączenia z kodem strzeżonym przez ograniczenia praw własności.

Przez kilka następnych lat Bostic i inni pracownicy uniwersytetu pracowali nad dorobieniem brakujących elementów kodu BSD, aby uczynić go pełnym, ogólnodostępnym systemem operacyjnym. Ich wysiłki ostatecznie zostały uwieńczone powodzeniem na początku lat dziewięćdziesiątych, pomimo opóźnień spowodowanych zastrzeżeniami prawnymi Unix Systems Laboratories z AT&T, które starał się zachować wyłącznie dla siebie prawa do używania nazwy firmowej Unix. Jednakże już wcześniej wiele narzędzi z Berkeley znalazło swe miejsce w projekcie GNU Stallmana.

"Wydaje się mało prawdopodobne, byśmy tak twardo parli do

przodu, gdyby nie wpływ projektu GNU - przyznaje Bostic, patrząc wstecz. - Było coś, co nas ostro pogańiało, a myśmy się temu z ochotą poddawali".

Do końca roku 1980 GPL pokazała swą moc przyciągania całej społeczności twórców oprogramowania. Narzędzia BSD należały do wolnego oprogramowania nawet bez licencji GPL, ale jej użycie niosło określony komunikat. "Myślę, że pojawienie się GPL zmusiło wielu ludzi do zastanowienia się, czy naprawdę tworzą wolne oprogramowanie i w jaki sposób powinni je licencjonować" - twierdzi Bruce Perens, twórca popularnego narzędzia uniksowego Electric Fence i przyszły grupy programistów Debian GNU/Linux. W kilka lat po pojawieniu się GPL postanowił zrezygnować z licencji własnego pomysłu i objąć Electric Fence licencją sporządzoną przez Stallmana przy wsparciu prawników. "Było to bardzo łatwe" - wspomina.

Rich Morin, który pierwszy anons Stallmana dotyczący projektu GNU przyjął sceptycznie, przyznaje, że z coraz większym zdumieniem spoglądał na gromadzenie się oprogramowania pod ochroną GPL. Jako przywódca grupy programistów SunOS w latach osiemdziesiątych był zobowiązany do rozsyłania taśm z nowymi wersjami najlepszego wolnego oprogramowania. Często wymagał o telefonowania do twórców oryginalnych programów i sprawdzania, czy przypadkiem prawa autorskie nie zabraniają ich rozpowszechniania i czy są przeznaczone do użytku w wolnej ogólnodostępnej domenie. Pod koniec roku 1989 zauważył, że najlepsze programy zwykle są chronione przez licencję GPL. "Jako dystrybutor oprogramowania wiedziałem, że znak GPL oznacza swobodę rozpowszechniania" - wspomina.

Aby zrekompensować sobie koszty wynikające z trudności kompletowania taśm z żądanymi programami, obciążał odbiorców stosowną opłatą. Teraz zauważył, że przygotowanie taśm złożonych z programów z licencją GPL zajmuje mu dwakroć mniej czasu, dzięki czemu interes zaczął przynosić zyski. Widząc okazję do zarobku, zamienił hobby w biznes pod

nazwą: Prime Time Freeware.

Taka działalność gospodarcza była całkowicie zgodna z zasadami wolnego oprogramowania. "Gdy mówimy o wolnym oprogramowaniu, mówimy o wolności, a nie o cenie" - objaśnia Stallman w preambule GPL. Pod koniec lat osiemdziesiątych przedstawił tę myśl w postaci uproszczonego stwierdzenia: "Mówiąc 'wolny', nie myśl o wolnym, bezpłatnym dostępie do piwa, lecz o wolności wypowiedzi".

Większość firm zlekceważyła argumenty Stallmana. Części przedsiębiorców wolność związana z wolnym oprogramowaniem nadal kojarzyła się z wolnym rynkiem. Usunięcie ograniczeń własności oprogramowania stwarzało sytuację, w której nawet najmniejsza firma mogła stawiać w szranki z takimi potęgami jak IBM i DEC.

Jednym z pierwszych przedsiębiorców, którzy to pojęli, był Michael Tiemann, programista i student ostatnich lat Stanford University. W latach osiemdziesiątych bez zastanowienia postępował zgodnie z zasadami projektu GNU, jak początkujący jazzman śladami swego ulubionego artysty. Dopiero po ukazaniu się w roku 1987 kompilatora GNU języka C (GNU C Compiler, w skrócie GCC) pojawił się wielki potencjał wolnego oprogramowania. Samo pojawienie się GCC, który nazwał "bombą", uznał za dowód wielkiej determinacji Stallmana-programisty.

"Tak jak każdy pisarz marzy o napisaniu wielkiej amerykańskiej powieści, tak każdy programista w latach osiemdziesiątych pragnął napisać wielki amerykański kompilator - wspomina Tiemann. - Nagle Stallman to zrobił. To był opanowanie".

"Mówicie o różnych niedociągnięciach GCC - wtóruje mu Bostic, - ale nikt przed Stallmanem nie mógł się uznać za twórcę kompilatora".

Zamiast współzawodniczyć ze Stallmanem, Tiemann postanowił korzystać z jego pracy. Oryginalna wersja GCC liczyła około 110 tysięcy linii kodu, ale Tiemann wspomina, że był to kod,

który można było bardzo łatwo odczytać i pojąć. Było to na tyle proste, że zajęło mu tylko pięć dni. Przez następny tydzień udało mu się przenieść to oprogramowanie na platformę mikroprocesora National Semiconductor 32032. W rok później zajął się tworzeniem kompilatora dla języka C+. Pewnego razu, wygłaszając na ten temat wykład w Bell Labs, zetknął się ze zmagającymi się z tym samym problemem programistami z AT&T.

"Na sali było około 40 lub 50 osób, więc zapytałem, ilu z nich pracuje nad tym kompilatorem - wspomina. - Gospodarz odpowiedział, że ta informacja jest objęta tajemnicą, ale mogę się mniej więcej zorientować, rozglądając się po sali".

Wkrótce potem, jak wspomina, rozjaśniło mu się w głowie.

"Pracuję nad tym kodem od sześciu miesięcy - pomyślał, - co oznacza niezłomną wydajność, którą wolny rynek powinien wynagrodzić".

Dalszą zachętę znalazł w manifestie GNU, gdzie obnażano chciwość niektórych dostawców oprogramowania, zachęcając innych do rozważenia korzyści, jakie wolne oprogramowanie daje konsumentom. Wolne oprogramowanie, łamiąc potęgę monopolu oprogramowania komercyjnego, umożliwia szybszym mniejszym dostawcom konkurowanie z wielkimi za pomocą usług i konsultacji - dwóch najbardziej dochodowych nisz rynku komputerowego.

W szkicu opublikowanym w roku 1999 Tiemann wspomina wrażenie, jakie wywarł na nim manifest Stallmana. "Czytało się to jak socjalistyczną polemikę, ale ujrzałem w tym coś zupełnie innego. Zobaczyłem tam zamaskowany biznesplan"¹².

Wraz z Johnem Gilmorem, innym zwolennikiem projektu GNU, założył firmę udzielającą konsultacji w dziedzinie korzystania i dostosowywania do potrzeb oprogramowania GNU. Cygnus Support, bo tak nazwano firmę, pierwszy kontrakt podpisał w lutym 1990 roku, a do końca roku miał już portfel kontraktów wart 725 000 dolarów.

GNU Emacs, GDB i GCC to "wielka trójka" narzędzi dla

programistów, ale nie był yto jedyne programy stworzone przez Stallmana w pierwszej połowie dekady istnienia projektu GNU. Do roku 1999 stworzył on wersje GNU Bourne Shell (przemianowanego na Bourne Again Shell czyli BASH¹⁴), YACC¹⁵ (przemianowanego na Bison), awk (przemianowanego na gawk¹⁶). Każdy program GNU na wzór GCC projektowano tak, aby mógł być uruchamiany w różnych systemach, a nie tylko na platformie jednego dostawcy. Podczas udoskonalania elastyczności i wszechstronności programów Stallman i jego współpracownicy często zwiększali ich przydatność.

Przypominając uniwersalistyczną filozofię GNU, Morin z Prime Time Freeware wskazuje na istotny i rozpowszechniony na świecie pakiet hello world¹⁷. "Program hello world liczy wszystkiego pięć linii kodu napisanego w C, ale jest opakowany, jakby był normalną dystrybucją GNU - mówi Morin. - Ma więc Textinfo (informacyjny plik tekstowy) i dane do konfiguracji. Dostał wszystkie dodatki inżynierii oprogramowania, jak wszystkie normalne pakiety GNU, dzięki czemu tak łatwo je instalować i uruchamiać na tak wielu różnych platformach. To diabelnie ważne i jest cechą nie tylko oprogramowania pisanego przez Stallmana lecz wszystkich pakietów tworzonych przez projekt GNU.

Według Stallmana, ulepszanie oprogramowania to zadanie wtórne, zaś najważniejsze było budowanie nowych programów. "Każdy kawałek programu daje, bądź nie daje się ulepszyć - mówił Stallman w wywiadzie dla Byte'a. - Do pewnego stopnia korzystam z reimplementacji, która prowadzi do ulepszania programów. W pewnej dużej mierze wynika to z faktu, że siedzę w tej branży tak długo, że pracowałem z wieloma systemami. Z tego powodu mam wiele pomysłów do zastosowania"¹⁸.

Do końca lat osiemdziesiątych narzędzia GNU zdążyły wyrobić sobie dobrą markę, a znana jeszcze z czasów AI Lab wybredność Stallmana stała się wprost legendarna w całym programistycznym świecie.

Jeremy Alson, w latach osiemdziesiątych użytkownik Suna i

programista realizujący własny projekt wolnego oprogramowania Samba, na początku lat dziewięćdziesiątych wspominał tę reputację ze śmiechem. Pod koniec lat osiemdziesiątych zaczął używać Emacsa. Zainspirowany rozwojem społeczności programistów, przesłał Stallmanowi fragment kodu jedynie po to, by został on odrzucony.

"To coś w rodzaju nagłówka z "The Onion"¹⁹ - mówi Allison. - "Modlącym się dzieciom Bóg odpowiedział: Nie!"

Rosnącemu uznaniu dla Stallmana programisty towarzyszyły kłopoty związane z zarządzaniem projektem GNU. Projekt odnosił jeden sukces za drugim w dziedzinie tworzenia narzędzi programistycznych, ale nie był w stanie stworzyć działającego jądra systemu, "centrali dowodzenia" wszystkich systemów uniksowych, decydującego o tym, które programy i narzędzia w danym momencie uzyskują dostęp do mikroprocesora. Zaczęło to wywoływać niepokój już pod koniec lat osiemdziesiątych. Jak w przypadku wielu innych zadań wykonywanych w projekcie GNU, Stallman rozpoczął pracę nad jądrem od poszukiwania istniejącego programu, który można by było zmodyfikować. Według informacji z "Gnusletters" ze stycznia 1987, przeglądał gruntownie TRIX, jądro uniksowe stworzone przez MIT.

Przeglądając "GNUsletters", biuletyn informacyjny projektu GNU z końca lat osiemdziesiątych, wyczuwa się rosnące napięcie. W styczniu 1987 Stallman poinformował, że w ramach projektu GNU dokonywana jest dokładna analiza TRIX-a, uniksowego jądra stworzonego przez MIT. Rok później, w lutym 1988 roku, podano do wiadomości, że w centrum zainteresowania znalazł się Mach, lekkie "mikrojądro" opracowane przez Carnegie Mellon. Wszyscy przyznają jednak, że właściwe prace nad jądrem nie rozpoczęły się przed rokiem 1990.

Opóźnienie prac nad jądrem było tylko jednym z problemów nękających wówczas Stallmana. W roku 1989 firma Lotus Development Corporation wystąpił przeciwko swojemu rywalowi na rynku programistycznym, firmie Paperback

Software International, oskarżając ją o skopiowanie poleceń menu z popularnego programu arkusza kalkulacyjnego 1-2-3. Nastąpił oto w tym samym czasie, gdy Apple i Microsoft toczył y walkę o "look and feel"²¹. Razem wzięte przyczynił o się to do powstania dodatkowych kłopotów odczuwanych przy realizacji projektu GNU. Choć oba procesy w żadnej mierze nie dotyczyły y projektu GNU, gdyż oba obracały się wokół systemów operacyjnych i aplikacji tworzonych dla komputerów osobistych niezgodnych sprzętowo z systemami uniksowymi, budził y jednak strach wśród wszystkich twórców oprogramowania. Stallman postanowił temu przeciwdziałać więc wraz z kilkoma zaprzyjaźnionymi programistami umieścił w jednym z magazynów ogłoszenie uderzające w trwające procesy. W ślad za tym zorganizował grupę zwalczającą te zjawiska. Przyjęła ona nazwę League of Programming Freedom²² i rozpoczęła protesty przed siedzibą Lotus, Inc. i przed budynkiem sądu w Bostonie, w którym toczył się rozprawa. Protest był zauważalny²³. To, co się działo, było odbiciem ewolucji przemysłu programistycznego, w którym aplikacje, a nie sam system operacyjny, stał y się gł ównym przedmiotem sporów. Niespeł nione dążenie GNU do stworzenia wolnego systemu operacyjnego zdawał się być anachronizmem. Rzeczywiście, fakt, że Stallman za nieodzowne uznał zebranie grupy zwalczającej procesy "look and feel", wielu odebrało jako oznakę starzenia się.

W roku 1990 Fundacja Johna D. i Catherine T. MacArthurów uznał a Stalmana za geniusza, nadając mu swoje członkostwo, co było połączone z wypłaceniem jego organizacji nagrody, tzw. "genius grant". Była to suma 240 tysięcy dolarów przeznaczona na rozwój projektu GNU. Projekt i głośzona idea wolnego oprogramowania na krótko stał y się wówczas obiektem powszechnego zainteresowania. Przede wszystkim pozwolił o to Stallmanowi i wielu jego nieopłacanym współpracownikom z Fundacji Wolnego Oprogramowania, dotychczas dorabiającym różnymi kontraktami konsultingowymi, poświęcić więcej czasu

na pisanie kodów dla projektu GNU.

Zabawne, że ta nagroda umożliwił a Stalmanowi także udział w głosowaniu. Kilka miesięcy przed jej przyznaniem pożar w mieszkaniu strawił niewiele rzeczy, które posiadał. Do czasu przyznania nagrody mieszkał jako "dziki lokator"²⁴ na Technology Square 545. "Komisarz wyborczy za nic nie chciał uznać, że taki jest mój adres - wspomina. - Pogodził się z tym dopiero wtedy, gdy ta informacja została potwierdzona w artykule prasowym o nagrodzie MacArthura."²⁵

Najważniejsze, że nagroda MacArthura dała Stallmanowi więcej swobody. Poświęcony bez reszty idei wolności oprogramowania, wykorzystał otrzymaną wolność na podróże wspierające misję projektu GNU.

Ciekawe, że jedna z tych podróży rzeczywiście przyczynił się do ostatecznego sukcesu projektu GNU i ruchu wolnego oprogramowania. W roku 1990 Stallman odwiedził Politechnikę w Helsinkach w Finlandii. Wśród jego słuchaczy znalazł się 21-letni Linus Torvalds, późniejszy twórca jądra Linuksa, elementu, który wypełnił dotkliwą lukę w oprogramowaniu projektu GNU.

Torvalds, wówczas student pobliskiego Uniwersytetu Helsińskiego, spoglądał na Stallmana bez entuzjazmu.

"Zobaczyłem po raz pierwszy w życiu typowego długowłosego i brodatego hakera" - wspomina w swojej autobiografii "Just for Fun" wydanej w roku 2001²⁶ i ²⁷.

Choć nie całkiem odpowiadała mu "socjopolityczna" strona zadań wyliczonych przez Stallmana²⁸, doceniał logikę zasadniczego założenia, że żaden programista nie pisze bezbłędnych kodów, a udostępnianie ich innym oznacza stawianie postępu ponad indywidualną żądzę zysku i chronienia własnego ego.²⁹

Jak wielu programistów z jego pokolenia, Torvalds nie zajął się problemami wielkich komputerów "mainframe", lecz "zatopił zęby" w różnobarwnej i niejednorodnej materii systemów komputerów osobistych. Jako student i użytkownik

uniwersyteckiego MicroVAXa, uczynił pierwszy krok, przechodząc od programowania w C do zagadnień systemu Unix. To wykonanie jednego kroku do przodu pozwoliło mu z innej perspektywy zobaczyć bariery broniące dostępu do maszyny. Dla Stallmana główną przeszkodą był ybiurokracja i przywileje, zaś dla Torvaldsa - geografia i ostra helsińska zima. Zmuszony brnąć przez teren uniwersytetu po to, by zalogować się do systemu uniksowego, rychło zaczął szukać możliwości łączenia się systemem bez wysuwania nosa ze swej ciepłej kwatery w uczelnianym kampusie.

Te poszukiwania doprowadziły go do systemu Minix³⁰, wersji "wagi lekkiej" Uniksa napisanej w celach edukacyjnych przez profesora holenderskiego uniwersytetu Andrew Tanenbauma. System pasował do ograniczonych parametrów, zwłaszcza niewielkiej pamięci PC 386, najpotężniejszej maszyny, na jaką Torvalds mógł sobie pozwolić, ale brakowało mu niektórych funkcji. Przede wszystkim nie pozwalał na emulację terminalu, co pozwalałoby na pełnienie roli terminalu uniwersyteckiego i logowanie się do MicroVAX-a z domu.

Latem 1991 roku Torvalds przetworzył od podstaw kod Minixa, dodając brakujące funkcje. Pod koniec lata zaczął określać swą pracę jako "terminalowe programy emulacyjne GNU/Emacs"³¹. Poczuli się wystarczająco pewni siebie, by poprosić miniksową grupę dyskusyjną o kopie standardów POSIX³², zbioru podstawowych zasad programistycznych, decydujących o tym, czy program można uznać za zgodny z Uniksem. Kilka tygodni później wysłał komunikat, bardzo przypominający stallmanowską zapowiedź powstania GNU w roku 1983:

Pozdrawiam wszystkich, którzy używają minixa
Robię (wolny) system operacyjny (jako hobby, więc nie będzie wielki i profesjonalny jak gnu dla klonów 386 (486) AT).
Szykuję to od kwietnia i zbliżam się do końca. Czekam na odzew o tym, co ludzie lubią/nie lubią w minixie, który mój system przypomina (fizycznie ten sam układ systemu plików (z

przyczyn praktycznych) i podobne inne rzeczy)³³.

Komunikat spotkał się z niewielkim odzewem, a w ciągu miesiąca Torvalds opublikował wersję 0.01 swego systemu operacyjnego, co oznacza, że został ona udostępniona na serwerze FTP³⁴. Stanął o przed nim zadanie nadania systemowi nazwy. Na dysku swojego własnego PC-ta zapisał ten program³⁵ pod nazwą Linux, zgodną z zasadą, że nazwy różnych wariantów Uniksa powinny się kończyć na X. Uznając ją jednak za "egoistyczną", zmienił ją na FreaX - chyba tylko po to, by administrator serwera FTP mógł dokonać zmiany odwrotnej.

Choć Torvalds założył zbudowanie pełnego systemu operacyjnego, zarówno on, jak i inni programiści doskonale wiedzieli, że wiele narzędzi jest już gotowych dzięki pracy GNU, BSD i innych programistów tworzących wolne oprogramowanie. Jednym z pierwszych narzędzi, z których skorzystała grupa twórców Linuksa, był GCC (GNU C Compiler), narzędzie pozwalające na przystosowanie do działania programów napisanych w języku C.

Integracja z GCC zwiększyła wydajność i możliwości Linuksa, co podniosło jego rangę. Choć "wirusowa" natura GPL nie miała zastosowania do jądra Linuksa, Torvalds, pragnąc pożyczyć GCC do własnych celów, zaciągnął zobowiązanie zwrócenia długu. Potem ujął to następująco: "Wzniosłem się na barkach olbrzymów"³⁶ i ³⁷. Nie zaskakuje, że zaczął rozmyślać, co by się stało, gdyby inni oczekiwali takiej samej pomocy. W dziesięć lat później zgodził się ze słowami Roberta Chassela z Fundacji Wolnego oprogramowania, który w następujący sposób podsumował jego ówczesne myśli:

Na zrobienie tego poświęciłem sześć miesięcy życia i chcę to udostępnić innym i dostać coś w zamian, ale nie chcę, aby inni czerpali z tego dla własnych korzyści. Chciałem, aby ludzie zobaczyli Linuksa, aby go zmieniali i ulepszali dla własnej satysfakcji, ale jednocześnie chciałem mieć dostęp do źródeł, aby móc robić te same ulepszenia, co oni³⁸.

Gdy nadszedł czas wydania wersji Linux 0.12, pierwszej w pełni zintegrowanej z GCC, Torvalds postanowił zgłosić przystąpienie do ruchu wolnego oprogramowania. Odrzucił starą licencję jądra systemu i zastąpił ją przez GPL, co zaowocowało potokiem adaptacji i Torvalds wraz ze współpracownikami mógł się przyglądać, jak wiele programów GNU wpada do kotła z "linuksowym gulaszem". W ciągu trzech lat linuksowi programiści mogli już pokazać Linuksa 1.0, zawierającego pełną i zmodyfikowane wersje GCC i GDB, oraz wiele narzędzi zaczerpniętych z BSD.

Do roku 1994 ten amalgamatyczny system operacyjny zdobył sobie już taki szacunek w hakerskim świecie, że niektórzy obserwatorzy zaczęli się interesować, czy Torvalds, zgadzając się na GPL, nie oddał "za frajer" kury znoszącej złote jaja. W pierwszym wydaniu Linux Journal wydawca Robert Young przeprowadził z nim wywiad. Gdy spytał fińskiego programistę, czy żałuje, że zrzekł się prawa własności do kodu źródłowego Linuksa, Torvalds zaprzeczył: "Nawet patrząc wstecz, na 100% uważam przyjęcie GPL za jedną z najlepszych decyzji, jakie podjąłem we wczesnej fazie projektu".³⁹

Fakt, że decyzja ta została podjęta bez odwoływania się do Stallmana i Fundacji Wolnego oprogramowania, świadczy o rosnącej przenośności GPL. Dopiero po paru latach Stallman uświadomił sobie, że siłą rozprzestrzeniania się Linuks przypomina zdarzenia dotyczące Emacsa. Tym razem przyczyną nie był wysztuczki hakerskie w rodzaju Ctrl+R lecz nowość, jaką był oddziaływanie systemu uniksowego na komputerach o architekturze PC-tów. Motywy działania były różne, ale rezultat końcowy był zgodny z założeniami etycznymi: powstał wolny, w pełni funkcjonalny system operacyjny.

Jak zapowiadał pierwszy e-mailowy komunikat wysłany przed kilkoma miesiącami do grupy comp.os.minix, Torvalds spodziewał się, że po ukończeniu przez programistów GNU jądra HURD⁴⁰ jego system rychło stanie się przeżytkiem. Ta początkowa niechęć spojrzenia na Linuksa z punktu widzenia

korzyści politycznych, okazała się ciosem, który Fundacja Wolnego Oprogramowania sama sobie wymierzył a. Jak zauważa Torvalds, był on tylko jednym z wielu dzieciaków, które gdzieś tam z boku robił różne rzeczy dla zabawy. Pomimo to, patrząc wstecz na oszałamiający sukces projektu, uznaje mądrość swej młodościowej decyzji zaakceptowania GPL, bez czego jego system mógł zakończyć żywot na jakimś zapomnianym przez wszystkich dysku twardym.

"A mogłem tego światła nie spostrzec - pisze, wspominając wykład Stallmana na Politechnice Helsińskiej w roku 1991 i wynikającą z tego decyzję przyjęcia GPL. - Widocznie jednak coś wtedy zapadło w moją duszę"⁴¹.

Przypisy

- 1 Patrz przypis 6. w rozdziale 2. - przyp. tłum.
 - 2 Patrz: Hal Abelson, Mike Fischer i Joanne Costello, Software and Copyright Law, wersja zaktualizowana (1998), <http://www.swiss.ai.mit.edu/6805/articles/int-prop/software-copyright.html>
 - 3 Patrz: plik README dołączony do Trn Kit: <http://www.za.debian.org/doc/trn/trn-readme>
 - 4 Richard Stallman, et al., GNU General Public License: Version 1, (luty 1989). <http://www.gnu.org/copyleft/copying-1.0.html>
 - 5 Ibidem.
 - 6 Tłumaczenie częściowo na podstawie polskiej wersji obecnej preambuły GNU GPL <http://www.gnu.org.pl/text/licencja-gnu.html> - przyp. tłum.
 - 7 Patrz: David Betz i Jon Edwards, Richard Stallman discusses his public-domain [sic] Unix-compatible software system with BYTE editors, BYTE (lipiec 1996). (Reprint na stronie projektu GNU: <http://www.gnu.org/gnu/byte-interview.html>).
- Ten wywiad pokazuje interesujące i szczere spojrzenie na

polityczne poglądy Stallmana we wczesnym okresie projektu GNU. Ułatwia także śledzenie zmian retoryki Stallmana. Stallman następująco opisuje cel stworzenia GPL: "Próbuję zmienić generalne podejście ludzi do wiedzy i informacji. Uważam, że próby zawłaszczenia wiedzy i nadzorowania dostępu do niej, zezwalanie lub zabranianie korzystania z niej lub dzielenia się nią z innymi - są sabotażem".

Warto to zestawić z wypowiedzią w trakcie rozmowy z autorem w sierpniu 2000: "Nalegam, aby nawet w myślach nie używać terminu Własność intelektualna. Prowadzi to do niezrozumienia rzeczy, gdyż określenie to ujednolica prawa autorskie (copyright), patenty i znaki handlowe. A są to różne rzeczy, niosące różne skutki i wspólne omawianie ich jest ogłupiające. Jeżeli słyszysz kogoś mówiącego bez cudzysłowu o własności intelektualnej, oznacza to, że nie myśli on wystarczająco jasno i nie ma sensu przyłączać się do niego".

8 Trudna do przetłumaczenia gra słowna, right to prawy, lecz także prawo, zasada prawna, zaś left - lewy, lewo, lecz również forma bierna czasownika leave, co głównie oznacza "zostawić, pozostawić", ale w szczególnym kontekście także "zezwoić". Skoro copyright to prawo autorskie ograniczające możliwości rozpowszechniania, to copyleft ma oznaczać zasadę umożliwiającą rozpowszechnianie, co poświadcza dalsza część tekstu na nalepce: "All Right Reversed" (wszystkie prawa odwrócone), co także jest przeróbką oryginalnego tekstu noty copyright - "All Rights Reserved" (wszystkie prawa zastrzeżone) - przyp. tłum.

9 Na polskiej stronie Fundacji Wolnego Oprogramowania czytamy: "Copyleft to ogólny sposób na nadanie programowi statusu wolnego oprogramowania i nakazanie, by wszystkie jego zmienione i rozszerzone wersje były również wolnym oprogramowaniem." Więcej informacji na ten temat patrz: <http://www.gnu.org/copyleft/copyleft.pl.html#WhatIsCopyleft> - przyp. tłum.

10 Grupa badawcza systemów komputerowych - przyp. tłum.

11 "Nieznosna klauzula ogłoszeniowa" University of California po pewnym stała się przyczyną problemów. Zaczęła się na nie wzorować część hakerów, szukając licencji mniej restrykcyjnej niż GPL, zastępując jedynie nazwę uniwersytetu nazwą własnej instytucji. Ponieważ wiele programów należących do grupy wolnego oprogramowania często czerpało kod dziesiątków innych programów, w ogłoszeniach należało o uwzględniać dziesiątki firm. W roku 1999, po dziesięcioletnim nacisku Stallmana, Uniwersytet Kalifornijski zgodził się na usunięcie tej klauzuli. Patrz: The BSD License Problem pod adresem <http://www.gnu.org/philosophy/bsd.html>

12 Michael Tiemann, Future of Cygnus Solutions: An Entrepreneur's Account, Open Sources (O'Reilly & Associates, Inc., 1999) s. 139.

13 GNU Emacs, GNU Debugger i GNU C Compiler - przyp. tłum.

14 BASH - Bourne Again Shell - (w wolnym tłumaczeniu oznacza nowo narodzoną powłokę) i jest typowym przykładem gry słów w Uniksie. BASH jest następcą powłoki Bourne'a, która wraz z powłoką Korn i powłoką C jest jedną z trzech klasycznych powłok Uniksa. W Linuksie dostępne są wszystkie trzy powłoki, ale standardowo zazwyczaj ustawiana jest powłoka BASH. Interpreter poleceń powłoki jest interfejsem między użytkownikiem a jądrem systemu, co oznacza, że polecenia wpisywane z klawiatury zamienia na język zrozumiały dla jądra systemu - przyp. tłum.

15 Yet Another Compiler Compiler (jeszcze jeden kompilator kompilatorów) - generator parserów (analizatorów składniowych), patrz: <http://pl.wikipedia.org/wiki/Yacc> oraz <http://www.linuxgazette.com/issue87/ramankutty.html> Kompilatorami kompilatorów nazywa się narzędzia wspomagające pracę twórców kolejnego języka programowania. - przyp. tłum.

16 awk jest skrótem złożonym z pierwszych liter nazwisk twórców języka, a są to Alfred V. Aho, Brian W. Kernighan, and

Peter J. Weinberger, gawk to GNU-awk. awk jest interpretowanym językiem programowania, którego główną funkcją jest wyszukiwanie i przetwarzanie wzorców. Więcej informacji można znaleźć np. na stronie: <http://pl.wikipedia.org/wiki/Awk> - przyp. tł um.

17 Witaj świecie - przyp. tł um.

18 Patrz: Richard Stallman, BYTE (1986).

19 Na temat serwisu www The Onion znalazł em ciekawą informację. Aby nie być posądzonym o koloryzowanie, przytaczam w oryginale (patrz <http://www.brokennewz.com/reports/onion.asp>):

"The Onion Headline Generator (1/9/2003 - Broken Newz)
New York - The well-known formerly funny internet website, The Onion, has been taken over by a corporate conglomerate that is undergoing a severe financial crisis. Because of the underlying weakness of the economy, The Onion has fired its staff of writers and its content is now being produced by a computer program that randomly assembles words into headlines."

Nieźle rozwiązanie. Gdy brakuje pieniędzy, zwalnia się autorów, a teksty generuje komputerowo. Opisywane w książce zdarzenie miał o miejsce jeszcze przed zastąpieniem autorów przez generator tekstów, ale również wówczas serwis był zbiorem dziwnie dobieranych informacji mających służyć rozrywce - przyp. tł um.

20 Patrz: HURD History

<http://www.gnu.org/software/hurd/history.html>

21 Spójrz i pocuj, spójrz i wyczuj, w czym rzecz, co w części polskich słowników informatycznych jest przekładane jako "wygląd i obsługa" - przyp. tł um.

22 Liga wolności programowania - przyp. tł um

23 League of Programming Freedom twierdzi, że protest zauważono z powodu pierwszego heksadecymalnego protest songu:

1-2-3-4, toss the lawyers out the door;

5-6-7-8, innovate don't litigate;
9-A-B-C, 1-2-3 is not for me;
D-E-F-O, look and feel have got to go.
<http://lpf.ai.mit.edu/Links/prep.ai.mit.edu/demo.final.release>

Co można przełożyć ze stratą rymu jako:

1-2-3-4, wyrzucić prawników za drzwi;
5-6-7-8, wprowadzaj innowacje, a nie apelacje;
9-A-B-C, nie dla mnie 1-2-3;
D-E-F-O, "patrz i czuj" musi nastąpić.

Warto zaznaczyć, że 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E i F to cyfry heksadecymalnego, czyli szesnastkowego systemu liczbowego - przyp. uzupełniony przez tłum.

24 Patrz: Reuven Lerner, Stallman wins \$240,000 MacArthur award, MIT, The Tech (18 lipca 1990),
<http://the-tech.mit.edu/V110/N30/rms.30n.html>

25 Patrz: Michael Gross, Richard Stallman: High School Misfit, Symbol of Free Software, MacArthur-certified Genius (1999).

26 Patrz: Linus Torvalds and David Diamond, Just For Fun: The Story of an Accidentaly Revolutionary (HarperCollins Publishers, Inc., 2001), s. 58 - 59.

27 Wydanie polskie: Linus Torvalds, David Daimond, Po prostu dla zabawy. Historia przypadkowej rewolucji
Wydawnictwo: K.E. Liber, Marzec 2002 - przyp. tłum.

28 Swoj stosunek do Stallmana i jego dzieł aTorvalds opisuje w wywiadzie udzielonym Alessandro Rubiniemu dla "Linux Gazette". Oto tłumaczenie fragmentu bezpośrednio związanego z tematem:

"Alessandro: Jaka jest twoja opinia o pracy i filozofii Richarda Stallmana?

Linus: Nie lubię mieszania polityki z zagadnieniami technicznymi i w wielu sprawach nie zawsze zgadzam się z rms. Dla niego wiele zagadnień dotyczących oprogramowania zdaje się być sprawą niemal religijną, a ja jestem tu o wiele bardziej

pragmatyczny. W rezultacie obaj wiemy, że w wielu kwestiach się nie zgadzamy i usilnie staramy się nie współpracować zbyt blisko, gdyż nie dałoby to dobrych rezultatów.

Na podstawie tego, co powiedziałem, ktoś mógłby odnieść wrażenie, że nie lubię rms, lecz nie jest to całkowicie prawdziwe. Był on i jest nadal siłą napędową większości zadań realizowanych przez ruchy "wolnego oprogramowania" i "otwartych kodów" i bez niego świat był by uboższy, a jemu do działania jest potrzebne nastawienie religijne.

Chyba najlepiej powiedzieć, że jestem admiratorem jego dzieła, ale nie chciałem być nim samym, gdyż nasze sposoby widzenia świata różnią się."

Zainteresowanych całością odsyłam na stronę:

<http://www.lg.ps.pl/issue32/rubini.html> - przyp. tłum.

29 Postawę Torvaldsa dobrze charakteryzuje tzw. prawo Linusa, podstawa modelu open source. Zainspirowane przez Linusa, a ukute przez Erica S. Raymonda, brzmi: "Przy dostatecznie dużej liczbie patrzących oczu każdy błąd (w programie komputerowym) jest niegroźny" - przyp. tłum. na podstawie http://pl.wikipedia.org/wiki/Linus_Torvalds

30 Skrót od MiniUNIX - przyp. tłum.

31 Patrz: Linus Torvalds i David Diamond, Just For Fun: The Story of an Accidental Revolutionary (HarperCollins Publishers, Inc., 2001), s. 78.

32 POSIX (Portable Operating System Interface - przenośny interfejs systemu operacyjnego). Próba standaryzacji różnych odmian systemu UNIX dokonana przez IEEE, znana również po nazwę IEEE 1003. Rozpoczęta około roku 1985. POSIX standaryzuje interfejs programistyczny i użytkownika oraz właściwości powłoki.

IEEE (Institute of Electrical and Electronics Engineers, Inc. - Instytut Inżynierów Elektryków i Elektroników) - wielka międzynarodowa organizacja z siedzibą w USA skupiająca inżynierów elektryków i elektroników z całego świata. Strona internetowa polskiej sekcji IEEE znajduje się pod adresem

<http://www.ee.pw.edu.pl/ieee/>

- przyp. tłum. na podstawie <http://pl.wikipedia.org> oraz <http://www.ee.pw.edu.pl/ieee/>

33 Patrz: Linux 10th Anniversary,

<http://www.linux10.org/history/>

34 File Transfer Protocol (protokół transmisji plików) - internetowy standard przesyłania plików - przyp. tłum.

35 Choć każdy system operacyjny jest w zasadzie programem, zwykle nie stosuje się tej nazwy. Użyłem jej świadomie w ślad za autorem dla podkreślenia, że mówimy o zaczątkach systemu, o programie, który dopiero stawał się w pełni użytecznym systemem - przyp. tłum.

36 Patrz: Linus Torvalds i David Diamond, Just For Fun: The Story of an Accidental Revolutionary (HarperCollins Publishers, Inc., 2001), s. 96-97.

37 Ta wypowiedź przypomina inną, którą z niewielkimi zmianami przytaczam za

http://pl.wikipedia.org/wiki/Linus_Torvalds:

Torvalds unika rozgłosu i zasadniczo odmawia komentarzy dotyczących konkurencyjnego oprogramowania, takiego jak system operacyjny Windows firmy Microsoft. Czasem jednak występuje ostro przeciwko temu, co jest postrzegane jako FUD (patrz niżej) uprawiany przez twórców zamkniętego oprogramowania, takich jak Microsoft. Na przykład piastujący stanowisko Senior-VP w firmie Microsoft Craig Mundie krytykował oprogramowania open source za brak innowacyjności i niszczenie własności intelektualnej. W liście odpowiadającym na tę krytykę Torvalds pisał:

"Zastanawiam się, czy Mundie słyszał kiedykolwiek o Izaaku Newtonie. Jest on sławny nie tylko ze względu na stworzenie podstaw mechaniki (i początki teorii grawitacji, o czym pamięta większość ludzi, łącznie z historią o jabłku), ale również z powodu stwierdzenia:

Jeśli byłbym w stanie spojrzeć dalej, to tylko dlatego, że stałbym na ramionach olbrzymów.

Jestem skłonny słuchać raczej Newtona niż Mundiego. Choć umarł prawie trzysta lat temu, mniej zasmradza pokój."

FUD (Fear, Uncertainty, Doubt - Strach, Niepewność, Zwątpienie). Strategia osłabiania konkurencji przez podawanie nieprawdziwych lub niejasnych informacji o rynku i produktach. Zakłada, że nawet, jeśli ktoś wie, że informacje nie są prawdziwe, jego pewność zostanie zachwiana, co osłabi jego pozycję - przyp. tłum.

38 Patrz: Linus Torvalds i David Diamond, Just For Fun: The Story of an Accidental Revolutionary (HarperCollins Publishers, Inc., 2001), s. 94 - 95.

39 Patrz: Robert Young, Interview with Linus, the Author of Linux, Linux Journal (1 marca 1994), <http://www.linuxjournal.com/article.php?sid=2736>

40 Jądro Hurd w zasadzie powstał o i jest mniej więcej zgodne ze standardem POSIX, ale wiąże się z nim wiele problemów, m.in. słaba wydajność i przenośność. Planuje się użycie Hurda jako jądra systemów GNU, ale na razie wciąż taką funkcję pełni Linux - przyp. tłum.

41 Patrz: Linus Torvalds i David Diamond, Just For Fun: The Story of an Accidental Revolutionary (HarperCollins Publishers, Inc., 2001), s. 59.

10. GNU/Linux

Przed rokiem 1993 ruch wolnego oprogramowania znalazł się na rozdrożu. Optymiści twierdzili, że wszystkie znaki na niebie i ziemi wskazują na zwycięstwo kultury hakerskiej. Magazyny sieciowe, dziwaczne nowe publikacje z opowiadaniem o szyfrowaniu danych, Usenet i wolności oprogramowania zaczęły gwałtownie zniknąć z półek. Internet - jeszcze niedawno slangowe określenie używane tylko przez hakerów i naukowców - znalazło swe miejsce w powszechnie używanym słownictwie. Używał go nawet prezydent Clinton. Komputer osobisty, do niedawna jedynie zabawka hobbystów, stał się powszechnie

uznawanym narzędziem dającym nowemu pokoleniu użytkowników dostęp do stworzonego przez hakerów oprogramowania. A skoro projekt GNU nie dostarczył jeszcze pełnego, wolnego systemu operacyjnego, ciekawscy użytkownicy, czekając nań, zajęli się wypróbowywaniem Linuksa.

Jakby na to nie patrzeć, wiadomości były dobre i po dekadzie walki o przetrwanie hakerzy wraz z promowanymi przez siebie wartościami poczęli zdobywać akceptację społeczną. Ludzie zaczęli "w to wchodzić".

Czy aby na pewno? Pesymiści twierdzili, że każdy znak akceptacji niesie ze sobą kolejny problem. Nagle okazało się, że świetnie być hakerem, ale czy było to dobre dla kultury, która dotychczas rozwijała się w odosobnieniu? Oczywiście, Biały Dom wychwalał Internet, a nawet zarejestrował własną domenę `whitehouse.gov`, ale towarzyszyły temu kontakty z firmami, zwolennikami cenzury i wpływowymi osobistościami dążącymi do prawnego postawienia tamy kulturze internetowej "dzikiego zachodu". Oczywiście, PC-ty stawały się coraz potężniejsze, ale w wyniku opanowania ich rynku przez chipy Intelu znacznie wzmocniła się pozycja dostawców oprogramowania z prywatnymi prawami własności. Na każdego użytkownika, który wybierał wolny system Linux, przypadły setki lub tysiące innych, którzy zaczynali od uruchamiania komputera z preinstalowanym systemem Windows.

Do tego dochodziła osobliwa natura Linuksa. Nie ograniczony błędami projektowymi (jak GNU) i debatami prawniczymi (jak BSD), błyskawiczny rozwój Linuksa nie był zaplanowany, a sukces tak przypadkowy i nieoczekiwany, że programiści najbliżej zaznajomieni z kodem, nie wiedzieli, co z tym począć. Była to raczej składanka niż system operacyjny, album, do którego wklejono hity oprogramowania, wszystko - od GCC, GDB i glibc (nowej biblioteki C projektu GNU), przez X (unikсовy interfejs graniczny stworzony przez MIT Laboratory for Computer Science), po opracowane przez BSD narzędzia,

takie jak BIND (ang. Berkeley Internet Naming Daemon - pozwalający zastąpić liczbowe adresy IP przez łatwe do zapamiętania nazwy domen) i TCP/IP. Oczywiście, zwornikiem tej konstrukcji był ojadro Linuksa - przenicowane i doładowane jądro Minixa. Zamiast budować system od podstaw Torvalds i jego grupa postąpili zgodnie z sentencją Picassa, że "dobrzy artyści pożyczają, wielcy artyści kradną". Później Torvalds, wyjaśniając tajemnicę sukcesu, tak to określił: "Jestem z natury bardzo leniwy i lubię polegać na tym, co robią inni"¹. Takie lenistwo, choć godne pochwały z punktu widzenia efektywności, z powodów politycznych sprawiało kłopoty. Podkreślał obrak celów ideologicznych. W przeciwieństwie do twórców GNU, Torvalds nie budował systemu operacyjnego z powodu chęci dostarczenia narzędzia kolegom hakerom; zbudował coś, czego sam potrzebował. Jak Tomek Sawyer bielący płót Torvalds nie zajmował się wizją ogólną lecz raczej zatrudnianiem innych hakerów w celu szybszego osiągnięcia wyniku.

Dlaczego więc jemu i współpracującym z nim hakerom udało się to, na czym inni poległi, i czym właściwie jest Linux? Czy jest to wyraz filozofii wolnego oprogramowania ogłoszonej w manifeście GNU napisanym przez Stallmana? A może to zwykły y amalgamat świetnych narzędzi, które inni podobnie myślący również byliby w stanie zmontować w system?

Przed upływem roku 1993 rosnąca liczba użytkowników Linuksa zaczął a się skłaniać ku tej drugiej definicji i zaczął a tworzyć przeróżne opinie na jego temat. Znaleźli się nawet tacy, którzy ośmielili się swoje wariacje na temat systemu nieco inaczej opakować i przedstawić uniksowym aficionados² jako oddzielne "dystrybucje". Rezultaty, nawet przy dobrej woli, trzeba nazwać partaniną.

"Było to jeszcze przed pojawieniem się Red Hata i innych komercyjnych dystrybucji - wspomina Ian Murdock, w owym czasie student Purdue University³. - W każdym uniksowym piśmie można było znaleźć kilka ogłoszeń wielkości wizytówki

reklamujących "Linux". Często były to firmy "krzaki", które nie widziały nic złego w próbie dodania własnego fragmentu do ogólnej składanki".

Murdock, uniksowy programista, został pochłonięty przez Linuksa w chwili, gdy po raz pierwszy zainstalował go na swoim komputerze. "To była wspaniała zabawa - mówi. - Wciągnęło mnie to". Jego zapał przygasił nieco zalew marnie napisanych dodatków. Zdecydował, że najlepszym rozwiązaniem będzie skonstruowanie wersji Linuksa oczyszczonego ze wszystkiego, co do niego doczepiono, więc przystąpił do sporządzania listy najlepszych dostępnych narzędzi programistycznych z zamiarem włączenia ich do własnej dystrybucji. "Chciałem stworzyć coś wartego nazwy Linuksa" - powiada.

Aby wzbudzić zainteresowanie innych, ogłosił swe zamiary w Internecie, również w grupie usenetowej comp.os.linux. Jeden z pierwszych e-maili, które otrzymał, pochodził od rms@ai.mit.edu. Jako haker, natychmiast rozpoznał autora. Był to Richard M. Stallman, twórca projektu GNU i człowiek znany Murdockowi jako "haker hakerów". Widok e-maila od Stallmana w skrzynce własnej poczty elektronicznej zdziwił go. Dlaczego, u diabła, Stallman, tworzący własny system operacyjny, miałby się interesować faktem, że on postanowił zająć się Linuksem?

Otworzył e-mail.

"Stallman pisał, że Fundacja Wolnego Oprogramowania zaczyna dokładnie przyglądać się Linuksowi i także być może będzie zainteresowana stworzeniem z niego systemu. Z grubsza można powiedzieć, iż Stallman sądził, że nasze zamierzenia są zgodne z ich filozofią".

Ten komunikat był odbiciem dramatycznej sytuacji Stallmana. Aż do roku 1993 nie wtykał on nosa w zdarzenia zachodzące wewnątrz linuksowej społeczności. Prawdę mówiąc, rozmyślnie unikał renegackiego systemu operacyjnego od chwili, gdy po raz pierwszy pojawił się on na horyzoncie w roku 1991. Po otrzymaniu pierwszej wiadomości o uniksopodobnym systemie

operacyjnym działającym na PC-tach zbadanie sprawy polecił jednemu z przyjaciół. "Powiedział mi - mówi Stallman, - że to oprogramowanie jest naśladownictwem Systemu V, pośledniej wersji Uniksa. Dodał, że nie jest przenośne".

Raport przyjaciela był słuszny. Linuks, zbudowany specjalnie dla PC 386, wykorzystywał maksymalnie możliwości tej technicznie ubogiej maszyny. Jednakże przyjaciel zapomniał wspomnieć o tym, że wielką wartością Linuksa było jego istnienie jako jedynego systemu operacyjnego, który można było oswobodnie modyfikować. Innymi słowy, w ciągu następnych trzech lat, które Stallman spędził na wysłuchiowaniu raportów o błędach składanych przez zespół pracujący nad stworzeniem HURD-a, Torvalds wysunął się przed programistów, którzy mieli później zająć się adaptacją Linuksa na inne platformy sprzętowe.

Do roku 1993 brak działającego jądra systemu projektu GNU stał się przyczyną kłopotów nie tylko grupy ludzi bezpośrednio w projekt zaangażowanych, lecz także całego ruchu wolnego oprogramowania. W marcu 1993 magazyn "Wired" w artykule napisanym przez Simsona Garfinkela poinformował, że projekt GNU "zapada się w grzęzawisko" pomimo sukcesów w tworzeniu wielu świetnych narzędzi⁴. Tym, którzy tkwili wewnątrz projektu i związanej z nim organizacji pozarządowej - Fundacji Wolnego Oprogramowania sytuacja wydawała się jeszcze gorsza od przedstawionej w artykule Garfinkela. "Było jasne, przynajmniej dla mnie, że los uśmiechnął się do nas, dając Tokienkoń, czyli czas na wprowadzenia nowego systemu operacyjnego - twierdzi Chassel. - Gdy się to Tokienkoń zamknęło, ludzie zaczęli tracić zainteresowanie. Właśnie to się zdarzyło"⁵.

Wielokrotnie analizowano przyczyny trudności, które dotknęły projekt GNU w latach 1990-1993. Niektórzy winą obciążali Stallmana, ale - jak twierdzi Eric Raymond, niegdyś członek zespołu GNU Emacs, a później krytyk Stallmana - problem miał bardziej instytucjonalny charakter. "FSF stał a się arogancka -

uważa Raymond. - Odeszła od tworzenia gotowego do działania systemu operacyjnego w stronę badań nad systemem. Nawet gorzej, bo sądzili, że nikt spoza Fundacji nie może mieć wpływu na ich pracę".

Murdock, osoba mniej wtajemniczona w wewnętrzne sprawy projektu GNU, widzi rzecz łagodniej. "Myślę, że częściową przyczyną kłopotów była ich wygórowana ambicja - zbyt wiele wysiłków marnotrawili na poprawianie błędów" - twierdzi. W latach osiemdziesiątych i dziewięćdziesiątych mikrojądra były gorącym tematem. Nieszczęściem projektu GNU było przystąpienie do budowy własnego jądra właśnie w tym czasie. Spowodowało to wielkie obciążenie prac i konieczność usilnego "pedałowania wstecz" w celu pozbycia się zbędnego bagażu. Stallman podaje wiele przyczyn wyjaśniających opóźnienie. Procesowanie się Lotusa z Apple odciągało uwagę, do tego doszła czasowa niezdolność Stallmana do używania klawiatury, co utrudniało mu udzielanie pomocy zespołowi pracującemu nad HURD-em. Inną przyczyną była marna wymiana informacji między różnymi częściami projektu GNU. "Mieliśmy wiele kłopotów z zachowaniem możliwości debuggowania środowiska, a zespół GDB w tym czasie nie był zbyt chętny do współpracy". Jednakże przyznaje, że główną przyczyną był fakt, iż on i jego współpracownicy nie docenili trudności, jakie sprawiało rozszerzenie mikrojądra Mach do postaci pełnego jądra uniksowego.

"Myślałem: OK, część (Mach), która ma się dogadać z maszyną, już została zdebuggowana - mówił Stallman, wspominając kłopoty z HURD-em w przemówieniu z roku 2000. - Mając gotową głowę, powinniśmy resztę dorobić znacznie szybciej, ale okazało się, że debuggowanie tych asynchronicznych wielowątkowych programów było nie lada kłopotem.

Wyznaczone terminy mijały i bieg czasu mógł zniweczyć nasze wysiłki. W rezultacie wyglądało na to, że wersja testowa pojawi się dopiero po latach⁶.

Jak by się nie usprawiedliwiać, stworzenie jądra Linux przez

konkurencyjny zespół zrodziło napiętą sytuację. Oczywiście, jądro to pojawiło się z licencją GPL, ale - jak zauważa Murdock - chęć uznania Linuxa za podstawowy system wolnego oprogramowania był a niewielka. Do końca roku 1993 liczba jego użytkowników wzrosła od mniej więcej tuzina entuzjastów Miniksa do pułapu gdzieś między 20 a 100 tys.⁷ To, co kiedyś był ojedynie hobby, stało się rynkiem dojrzałym do eksploatacji. Gdy doszło do uznania zwycięstwa Linuksa, Stallmanem miotały mieszane uczucia - jak niegdyś Churchilllem patrzącym na wkraczanie do Berlina sowieckich oddziałów⁸.

Choć Stallman późno zasiadł do gry, ma w niej jednak swój udział. Gdy tylko FSF ogłosiła, że jest gotowa poprzeć finansowo i moralnie projekt Murdocka, zaczął napływać kolejne oferty wsparcia. Murdock nadał projektowi nazwę Debian - co było kompilacją imion jego żony i jego: Deborah i Ian - i już po kilku tygodniach przedstawił pierwszą gotową dystrybucję. "(Poparcie Richarda) niemal w ciągu jednej nocy przekształciło Debian z małego, interesującego niewiele osób, lokalnego projektu w przedsięwzięcie, na które niemal cała społeczność zwróciła uwagę" - twierdzi Murdock.

W styczniu 1994 roku Murdock wydał "Manifest Debian". Napisany w duchu stallmanowskiego, wydany przed dekadą, "Manifestu GNU", wyjaśnia wagę współpracy z Fundacją Wolnego Oprogramowania (FSF). Murdock pisze⁹:

Fundacja Free Software Foundation może odegrać niezmiernie ważną rolę w przyszłości Debiana. Już poprzez fakt udostępniania go przez tę organizację świat otrzyma sygnał, że Linux nie jest produktem komercyjnym i nigdy nie powinien nim zostać. Nie znaczy to, iż nigdy nie będzie mógł stanowić konkurencji na polu komercyjnym. Tym, którzy mają przeciwnie zdanie, polecam przyjrzenie się fenomenowi programów GNU Emacs albo GCC, które nie są aplikacjami komercyjnymi, a jednak wywołały spore zamieszanie na rynku podobnego oprogramowania.

Przyszedł czas bardziej skoncentrować się na przyszłości

Linuksa niż na nieetycznym celu, jakim jest wzbogacanie się kosztem całej linuksowej społeczności. Rozwój i dystrybucja Debiana niekoniecznie stanowią rozwiązanie problemów, których obraz naszkicowałem w manifestcie, ale mam nadzieję, że przynajmniej zwrócą na nie uwagę i w rezultacie doprowadzą do ich rozwiązania¹⁰.

Krótko po wydaniu manifestu Fundacja Wolnego Oprogramowania wysunął a swe pierwsze żądanie. Stallman chciał, aby Murdock nadał dystrybucji nazwę "GNU/Linux". Jak twierdzi Murdock, Stallman najpierw zaproponował nazwę Lignux, jako "Linux z GNU w sercu", ale krótkotrwałe próby użycia tej nazwy na listach Usenetu wywołały liczne drwiny hakerskiej braci, co skłoniło Stallmana do przyjęcia mniej niezręcznej nazwy GNU/Linux.

Choć niektórzy uznali żądanie Stallmana za niewczesne wyciąganie ręki po czyjaś chwałę, Murdock widział to inaczej. Siegając myślą wstecz, widzi to jako echo kontrowersji i wzrostu napięcia między projektem GNU i grupą twórców linuksowego jądra. "Wyraźnie było widać pojawiający się rozdźwięk - wspomina. - Richard przejął się tym".

Przyczyną najgłębszego podziału była glibc. glibc to skrót od GNU C Library (biblioteka GNU C), którym nazwano pakiet programów pozwalających programistom na kierowanie do jądra "wywołań systemowych" (wywołań usług systemowych). W latach 1993 - 1994 glibc był a "wąskim gardłem" Linuksa. Ponieważ wielu użytkowników dodawało do jądra swe własne funkcje, ludzie pracujący w projekcie GNU nad utrzymaniem glibc zostali wkrótce przytłoczeni nadmiarem żądań. Część użytkowników Linuksa, sfrustrowanych opóźnieniami i szerzącym się przekonaniem o rozłajności projektu GNU, domagała się "rozwidlenia" działań i stworzenia równoległej do glibc biblioteki C.

W świecie hakerskim rozwidlenia są ciekawym zjawiskiem. Choć zasady etyki hakerskiej zezwalają programistom na dowolne przerabianie w kodzie źródłowym wszystkiego, na co

mają ochotę, większość woli umieszczać swe zmiany w centralnym pliku źródłowym lub w "drzewie", aby zapewnić zgodność z programami tworzonymi przez innych. Rozwidlenie glibc na tak wczesnym etapie rozwoju systemu groziło potencjalną utratą setek, a być może tysięcy przyszłych linuksowych programistów. Groziło to także zwiększeniem niezgodności Linuksa i reszty systemu GNU, nad którą Stallman i jego grupa nadal pracowali.

Jako lider projektu GNU Stallman w roku 1991 już doświadczył negatywnych rezultatów rozwidlenia oprogramowania. Grupa twórców Emacsa pracująca dla programistycznej firmy Lucid przeszedł do porządku dziennego nad niechęcią Stallmana do uwzględniania części poprawek w podstawowym kodzie GNU Emacs. Doprowadziło to do powstania równoległej wersji Lucid Emacs oraz wielu kłopotów i starć¹¹.

Murdock twierdzi, że Debian wyrosło z takich samych chęci rozwidlenia kodu, które leżały u podstaw żądań dotyczących glibc. Dlatego właśnie Stallman domagał się umieszczenia części GNU, gdy tylko Debian stał o się dystrybucja zdolną do rozpowszechniania. "Dzięki temu rozwidlające się drogi zbiegły się ponownie, jednakże społeczność linuksowa nadal odczuwała swą odrębność w stosunku do społeczności GNU, co mogło stać się siłą niszczącą jedność".

Stallman potwierdza opinię Murdocka. Jak twierdzi, powstawaniu wszystkich najważniejszych elementów projektu GNU towarzyszył zagrożenie rozwidlenia. Mówi, że z początku uważał rozwidlenie za wynik rozczarowania, zawiedzenia nadziei. W przeciwieństwie do szybko działającej nieformalnej grupy twórców jądra Linuksa, pracujący dla GNU programiści starali się pracować wolniej, bez gwałtownego dostosowywania się do okoliczności, ale z myślą o przygotowaniu zmian, które będą miały znaczenie dla działania systemu w długim czasie. Nie obawiali się ostrej krytyki innych twórców kodu. Jednakże w miarę upływu czasu Stallman, czytając e-maile napływające od innych linuksowych programistów, wyczuł malejące

zainteresowanie projektem GNU i jego celami.

"Odkryliśmy, że ludzie uważający się za linuksowych programistów przestają się interesować projektem GNU - mówi Stallman. - Mówili 'Dlaczego mam się tym przejmować? Co mnie obchodzi projekt GNU? Pracują dla nas, użytkowników Linuksa, a poza tym, co mi do tego'. Zastanawiające było o, że nawet ludzie używający jakiegoś wariantu systemu GNU sam projekt GNU mieli w nosie."

Choć kilka wcześniejszych opisów Linuksa jako "wariantu" projektu GNU wydawał o musie chwytem politycznym, Murdock, sympatyzujący z racjami wolnego oprogramowania, uznał za uzasadnione żądanie Stallmana, by Debianowi nadać nazwę GNU/Linux. "Przystałem na to bardziej ze względu na jedność niż z powodu uznania za sług" - mówi.

Potem nastąpiły żądania natury bardziej technicznej. Choć Murdock przystosował się do żądań politycznych, usztywnił swą pozycję, gdy przyszło do ustalania zasad rozwoju istniejącego oprogramowania. To, co zaczęło się jako manifestacja solidarności, rychło stało się modelem dla innych projektów GNU.

"Muszę przyznać, że często się z nim nie zgadzałem - mówi Murdock ze śmiechem. - Z Richardem naprawdę trudno współpracować".

Po uzyskaniu dyplomu Purdue w roku 1996 Murdock postanowił przekazać innym sterowanie rozwijającym się projektem Debian. Obowiązki zarządcy powierzył Bruce'owi Perensowi, hakerowi znanemu z pracy przy Electric Fence, narzędziu uniksowym z licencją GPL. Perens, tak samo jak Murdock, był programistą uniksowym, który zakochał się w GNU/Linuksie, gdy tylko stały się widoczne jego uniksopodobne możliwości. Podobnie jak Murdock sympatyzował z politycznym przesłaniem Stallmana i Fundacji Wolnego Oprogramowania, jednak zachowywał pewien dystans.

"Pamiętam, że wkrótce po ogłoszeniu manifestu GNU,

wypuszczeniu GNU Emacsa i GCC wyczytał em w jakimś artykule prasowym, iż Stallman pracuje jako konsultant Intela - mówi Perens, wspominając swoje pierwsze starcie ze Stallmanem w roku 1980. - Napisałem do niego, pytając, w jaki sposób godzi rolę adwokata wolnego oprogramowania z funkcją konsultanta Intela. Odpowiedział : "Pracuję jako konsultant do spraw tworzenia wolnego oprogramowania." Odpowiedź była bardzo grzeczna i myślę, że sensowna".

Jednakże przyjmując na siebie obowiązki kierowania rozwojem Debiana, Perens uznał starcia Murdocka ze Stallmanem za kłopotliwe. Stając na czele zespołu, podjął strategiczną decyzję o oddzieleniu Debian od Fundacji Wolnego Oprogramowania. "Zdecydowałem, że nie możemy przystać na stallmanowski styl mikrozarządzania" - mówi.

Wedł ug Perensa, ta decyzja zaskoczył a Stallmana, ale był wystarczająco mądry, by przejść nad nią do porządku dziennego. "Odczekał nieco, by ochłonać i wysł ał e-mail, który był istotny dla ustalenia wzajemnych stosunków. Zażądał zachowania nazwy GNU/Linux, na co przystał em. Decyzję podjąłem samodzielnie. Wszyscy odetchnęli z ulgą."

Po pewnym czasie Debian zdobył opinię hakerskiej wersji Linuksa, podobnie jak Slackware, inna popularna dystrybucja powstał a w latach 1993-1994. Jednakże Linux zaczął zdobywać pozycję na komercyjnym rynku uniksowym. W Północnej Karolinie firma uniksowa o nazwie Red Hat przestawił a swoją działalność na linuxowe tory. Jej dyrektorem naczelnym był Robert Young, wcześniej wydawca Linux Journal¹², który w roku 1994 zapytał Linusa Torvaldsa, czy żałuje decyzji podporządkowania linuxowego jądra licencji GNU. Odpowiedź Torvaldsa znacznie wpł ynęł a na sposób postrzegania Linuksa przez Younga. Zamiast dążyć do zepchnięcia go w kąt rynku oprogramowania za pomocą tradycyjnie stosowanej taktyki, Young zaczął się zastanawiać nad zastosowaniem w firmie podejścia debianowskiego, budowania kompletnego systemu operacyjnego z całkowicie wolnych elementów. Cygnus

Solutions, firma założona w roku 1990 przez Michaela Tiemanna i Johna Gilmore'a pokazała już możliwość sprzedawania usług polegających na poprawianiu jakości i dostosowywaniu wolnego oprogramowania do potrzeb klientów. Co by się stało, gdyby Red Hat w ten sam sposób podszedł do Linuksa?

"W zachodniej tradycji naukowej my dzisiaj stoimy na ramionach gigantów - powtarza Young za Torvaldsem i Newtonem. - W języku biznesu mówimy, że koła nie należy wciąż odkrywać na nowo. O pięknie licencji GPL stanowi oddanie kodu domenie publicznej¹³. Jeżeli jesteś niezależnym dostawcą oprogramowania i do stworzenia jakiejś aplikacji jest ci potrzebny program wdzwanający się przez modem (modem dialer), zamiast pisać go od nowa możesz wziąć PPP z Red Hat Linuksa i użyć go jako narzędzia we własnym programie. Jeżeli potrzebujesz zestawu narzędzi graficznych, nie musisz pisać od nowa całej biblioteki graficznej. Po prostu ściągasz GTK¹⁴. Nagle masz możliwość korzystania z najlepszego, co dotychczas zostało zrobione. Jako dostawca aplikacji mniej czasu musisz poświęcać na zarządzanie oprogramowaniem i możesz go przeznaczyć na pisanie kodu aplikacji dostosowującej narzędzia do specyficznych potrzeb klienta.

Young nie był jedynym zarządcą firmy zainteresowanym możliwościami biznesowego wykorzystania wolnego oprogramowania. Do końca roku 1996 większość firm uniksowych przebudziła się, węsząc zapach "świeżo warzonego" kodu. Sektor linuksowy był dopiero dobre dwa, trzy lata przed pełnym komercyjnym przełomem, ale ci, którzy byli blisko hakerskiej społeczności, już "czuli pismo nosem", przewidywali wielkie wydarzenia. Chip 386 Intela, Internet i World Wide Web uderzył w rynek jak wielkie oceaniczne fale, ale największą okazał się Linux i korzystające z niego oprogramowanie wraz ze wspólną liberalną polityką licencyjną i wolnością korzystania z kodów źródłowych.

Dla Iana Murdocka, najpierw zwolennika Stallmana, później

odstręconego przez jego styl zarządzania, napływająca fala zdawała się należą nagrodą i należną karą dla człowieka, który znaczną część życia poświęcił na nadanie wolnemu oprogramowaniu właściwej tożsamości. Jak wielu linuksowych aficionados, Murdock widział, jak naprawdę wszystko się działo. Był świadkiem oryginalnego ostrzeżenia Torvaldsa, że Linux "jest tylko hobby", a także wyznania Torvaldsa pod adresem twórcy Miniksa Andrew Tanenbauma, że: "gdyby jądro GNU było tego lata gotowe, w ogóle nie przystępowałbym do realizacji swojego projektu"¹⁵. Tak jak inni Murdock widział możliwości, które zostały zmarnowane. Odczuwał także podniecenie, jakie niesło obserwowanie nowych możliwości niesionych przez każde nowe "okno" Internetu.

"Dzięki Linuksowi tkwiłem w tej zabawie od początku - wspomina. - Mieliliśmy wówczas coś do zrobienia, aby jakoś wykorzystać upływający czas. Jeżeli przeczytacie tamte stare dyskusje (comp.os.minix), wyczujecie sentymentalną nutę: robiliśmy coś, w co mogliśmy się bawić dopóty, dopóki nie pojawi się gotowy HURD. Ludzie byli podnieceni. To zabawne, ale podejrzewam, że gdyby HURD powstał wcześniej, Linux nigdy by się nie przydarzył".

Przed końcem roku 1996 pojawiły się kwestie sporne.

Użytkownicy dyskutowali, jak to nazwać - Linux, a może GNU/Linux. 36 miesięcy pozostawionych na stworzenie jądra dobiegł okresu, okno zostało zatrzaśnięte, co oznacza, że gdyby nawet projekt GNU zbudował wreszcie swoje jądro HURD, miałoby ono niewielkie szanse na zauważenie przez kogoś spoza ścisłego grona hakerskiej społeczności. Pierwszy uniksopodobny wolny system operacyjny już istniał i nabierał pędu. Hakerom pozostało jedynie oczekiwanie na kolejną falę, która miała anadziagnąć i spaść na ich głowy. Nawet na kudłatą głowę Richarda M. Stallmana.

Gotową na to lub nie.

Przypisy

1 Torvalds powtarzał to wielokrotnie. Najbardziej znane jest przytoczenie tych słów w eseju Erica Raymonda: The Cathedral and the Bazaar (maj 1997),

<http://www.tuxedo.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/index.html>

2 Aficionado (hiszp.) - zwolennik, miłośnik, oddany czemuś - przyp. tł um.

3 Uniwersytet w Lafayette w stanie Indiana, założony w roku 1869. Nazwę otrzymał dla upamiętnienia swego dobroczyńcy Johna Purdue, który jego założenie wsparł kwotą 150 tys.

dolarów, co stanowiło 3/4 całej potrzebnej sumy - przyp. tł um.

4 Patrz: Simson Garfinkel, Is Stallman Stalled? Wired (marzec 1993).

"Okienko", o którym mówi Chassel, to okres 36 miesięcy, ale takie kłopoty był udziałem nie tylko projektu GNU. We wczesnych latach dziewięćdziesiątych dystrybucja wolnych wersji Berkeley Software Distribution została wstrzymana przez ograniczenia prawne stworzone przez Unix System Laboratories. Choć wielu użytkowników uważało pochodne BSD, w rodzaju FreeBSD i OpenBSD, za górujące nad GNU/Linux zarówno pod względem wydajności, jak również bezpieczeństwa, liczba użytkowników tych systemów pozostaje jedynie ułamkiem populacji linuksowej.

Analizę względną sukcesu GNU/Linuksa i innych wolnych systemów operacyjnych znajdziemy w artykule nowozelandzkiego hakera Liama Greenwooda Why is Linux Successful (1999).

6 Patrz: wykład w Maui High Performance Computing Center.

7 Liczba użytkowników Linuksa jest znana tylko w zarysie, stąd tak duża rozbieżność. Liczba 100 tys. pochodzi ze strony "Milestones" serwisu Red Hat:

<http://www.redhat.com/about/corporate/milestones.html>.

8 Napisałem tę analogię o Churchillu, zanim Stallman

przysłał mi jej potwierdzenie:

Druga wojna światowa i determinacja potrzebna do zwycięstwa był ybardzo silnym wspomnieniem w latach mojego dzieciństwa. Wypowiedzi takie jak churchillowska zapowiedź "Będziemy z nimi walczyć na lądzie, będziemy z nimi walczyć na plażach... nigdy się nie poddamy" znajdowały żywy oddźwięk w mojej duszy.

9 Tłumaczenie to przytaczam z polskiej strony www projektu Debian: <http://www.debian.org/international/Polish/manifest> - przyp. tłum.

10 Patrz: Ian Murdock, A Brief History of Debian, (6 stycznia 1994): Dodatek A, "The Debian Manifesto".

<http://www.debian.org/doc/manuals/project-history/apA.html>

11 Informacje o rozwidleniu wersji Lucid/GNU Emacs można znaleźć na stronie WWW Jamiego Zawinskiego, niegdyś programisty w firmie Lucid, który potem stanął na czele grupy tworzącej Mozillę. Patrz: artykuł "The Lemacs/FSFmacs Schism" pod adresem <http://www.jwz.org/doc/lemacs.html>

12 Patrz: <http://www.linuxjournal.com/> - przyp. tłum.

13 Young niesłusznie używa tu terminu "domena publiczna". Domena publiczna nie jest chroniona przez prawa autorskie (copyright). GPL z definicji stanowi ochronę praw autorskich.

14 Informacje i różne wersje biblioteki do ściągnięcia na stronie <http://www.gtk.org/> - przyp. tłum.

15 Cytat pochodzi z "podrasowanej" do publikacji "płomiennej wojny" (ang. flame war) Torvaldsa z Tanenbaumem, która wybuchła po publikacji wstępnej wersji Linuksa. Wyjaśniając przyczyny zdecydowania się na stworzenie monolitycznego nieprzenoszalnego jądra systemu, Torvalds twierdzi, że rozpoczął pracę nad Linuksem, dążąc do dokł adniejszego poznania PC 386. "Gdyby jądro GNU było tamtej wiosny gotowe, w ogóle nie wpadłby mi do głowy pomysł rozpoczynania własnego projektu". Patrz: Chris DiBona et al., Open Sources (O'Reilly & Associates, Inc., 1999), s. 224.

11. OPROGRAMOWANIE OPEN SOURCE1

W listopadzie 1995 Peter Salus, członek Fundacji Wolnego Oprogramowania, autor książki z roku 1994 *A Quarter Century of Unix* (Ćwierć wieku Uniksa), przesłał do listy dyskusyjnej zajmującej się systemami projektu GNU prośbę o nadsyłanie artykułów konferencyjnych. Conference on Freely Redistributable Software (konferencja na temat swobodnie rozpowszechnianego oprogramowania) miała być zorganizowana w Cambridge w stanie Massachusetts w lutym 1996, a Salus, wybrany na jej przewodniczącego, chciał w ten sposób zawiadomić o tym kolegów hakerów. Sponsorowana przez Fundację Wolnego Oprogramowania, miała stać się pierwszą konferencją inżynierii oprogramowania w całości poświęconą wolnemu oprogramowaniu. Dla zademonstrowania jedności zaproszono także programistów z innych ruchów wolnego oprogramowania zajmujących się "różnymi aspektami GNU, Linuksa, NetBSD, 386BSD, FreeBSD, Perla, Tcl/tk oraz innych narzędzi, których kod był wolnodostępny i mógł być swobodnie rozpowszechniany". Salus pisał :

Przez ostatnie 15 lat wolne i tanie oprogramowanie stało się wszechobecne. Ta konferencja ma zebrać razem wszystkich, którzy dokonali implementacji różnego rodzaju wolnego oprogramowania i jego wydawców (na różnego rodzaju nośnikach). Przewidziane są wykłady informacyjne i artykuły odnoszące się do tematu, a także przewodnie wystąpienia Linusa Torvaldsa i Richarda Stallmana².

Jednym z pierwszych, do których dotarł e-mail Salusa, był Eric S. Raymond, jeden z członków komitetu tej konferencji. Choć nie kierował żadnym projektem, ani firmą, jak większość innych z listy, zdobył sobie przyzwoitą reputację w społeczności hakerskiej jako jeden z najaktywniej biorących udział w GNU Emacs i jako wydawca *The New Hacker Dictionary* (Nowego słownika hakerskiego), książkowej wersji liczącego się

już pliku słownikowego Jargon File.

Dla Raymonda konferencja w roku 1996 była oczekiwanym zdarzeniem. W latach osiemdziesiątych brał aktywny udział w projekcie GNU, po czym w roku 1992 usunął się na bok, zrażony - jak wielu przed nim - mikrozarządzaniem Stallmana. "Richard zrobił awanturę z powodu moich nieautoryzowanych modyfikacji, które wprowadziłem podczas czyszczenia emacsowych bibliotek lispowych - wspomina. - Tak mnie to sfrustrowało, że postanowił on dalej z nim nie współpracować". Pomimo swego odejścia Raymond pozostał członkiem społeczności wolnego oprogramowania. Na tyle aktywnym, by gorliwie przyklasnąć pomysłowi Salusa, gdy ten zaproponował, aby na konferencji obok siebie wystąpili Torvalds i Stallman. Stallman reprezentował stary zespół hakerski związany z ITS/Unix, zaś Torvalds - młodą i dynamiczną grupę hakerów linuksowych. Ich wystąpienia na tej samej konferencji symbolizowały yjedność, która mogła przynieść pożądane owoce zwł ąszcza dla młodych (tzn. poniżej czterdziestego roku życia) i ambitnych hakerów w rodzaju Raymonda, co on podsumował następująco "Poczułem, jakbym stał każdą nogą w innym obozie".

W okresie przed konferencją napięcie między tymi obydwoima obozami stawał o się coraz bardziej wyczuwalne. Łączył aje wspólna chęć zobaczenia w naturze fińskiego "cudownego dziecka". Ku zaskoczeniu wszystkich Torvalds okazał się przyjemnym, czarującym mówcą. Mówiąc jedynie z lekkim szwedzkim akcentem, zdumiał audytorium bł yskotliwym, zabarwionym autoironią dowcipem³. "Jeszcze bardziej zdumiewały jego cięte, rzucane mimochodem złośliwości pod adresem innych hakerów, nie wył ączając Stallmana" - twierdzi Raymond. Do końca konferencji wszystkich młodszych i starszych uczestników ujął jego lekki, pół hakerski, pół odmienny sposób bycia.

"Był to moment przełomowy - wspomina Raymond. - Przed rokiem 1996 Richard był jędnym uznanym przywódcą

ideologicznym całej społeczności. Dysydenci nie występowali publicznie. Torvalds był tym, który złamał to tabu".

Do następnego przełamania tabu doszło pod koniec konferencji. Podczas dyskusji o nadciągającej dominacji Microsoftu i jego systemu Windows Torvalds stwierdził, że podoba mu się PowerPoint, program Microsoftu do przygotowywania i pokazywania prezentacji. Dla hakerskich purystów ideologicznych było to jak chęłpienie się mormona w świątyni swym zamiłowaniem do whisky, zaś dla Torvaldsa i jego zwolenników była to po prostu opinia zgodna ze zdrowym rozsądkiem. Dlaczego unikać wartościowych programów komercyjnych? Celem hakera jest wykonanie zadania, a nie cierpienie dla idei.

"Było to szokujące stwierdzenie - wspomina Raymond. - Mógł sobie na nie pozwolić, gdyż w latach 1995 i 1996 gwałtownie zdobywał coraz szersze wpływy".

Stallman z kolei nie pamięta napięcia narastającego podczas konferencji w roku 1996, ale pozostała mu w pamięci impertynencja Torvaldsa. "W dokumentacji Linuksa znalazło się stwierdzenie, by wydrukować wszystkie standardy programistyczne GNU, a potem je podrzeć - przytacza jeden przykład. - Dobrze, mógł się nie zgadzać z naszymi konwencjami, ale nie musiał wyrażać tego w tak arogancki sposób. Czyż nie mógł powiedzieć po prostu 'Ja tak to widzę'? Nie byłoby w tym demonstracji wrogości".

Gorące przyjęcie Torvaldsa przez innych hakerów jedynie potwierdziło podejrzenia Raymonda. Główna linia oddzielająca twórców Linuksa i GNU/Linuksa zgadzała się z podziałem pokoleniowym. Wielu hakerów linuksowych, jak Torvalds, wyrosła w świecie oprogramowania komercyjnego z zastrzeżonymi prawami własności. Jeżeli program nie jest zły, dlaczego go unikać tylko z powodu rodzaju licencji? Gdzieś tam w przestrzeni wolnego oprogramowania kiełkował program, który być może kiedyś zostanie przekształcony w wolny odpowiednik PowerPointa. Ale póki się to nie stanie, dlaczego

zazdrościć Microsoftowi i zwalczać go?

Jako uczestnik projektu GNU z większym stażem Raymond wyczuwał rosnące napięcie między Stallmanem i Torvaldsem. W ciągu dekady, która upłynęła od rozpoczęcia projektu GNU, Stallman zdobył budzącą respekt reputację doskonałego programisty. Zdobył sobie również opinię człowieka nieprzejechanego zarówno w dziedzinie programowania, jak i zarządzania ludźmi. Krótco przed konferencją z roku 1996 Fundację Wolnego Oprogramowania opuścił cały personel zrażony postępowaniem Stallmana. Brian Youmans, obecny pracownik FSF, przyjęty przez Salusa po zbiorowym odejściu innych, wspomina, że "Wówczas Peter (Salus) był jedyną osobą pracującą w biurze".

Zbiorowe odejście pracowników jedynie potwierdziło podejrzenia Raymonda, że ostatnie opóźnienia, takie jak z HURD-em i ostatnie kłopoty, takie jak ze schizmą Lucid-Emacs wynikały nie z trudności w tworzeniu kodu, lecz z nieudolnego zarządzania projektem. Wkrótce po Freely Redistributable Software Conference Raymond rozpoczął prace nad swym własnym projektem-pupilką, narzędziem popmailowym (klienta pocztowego) o nazwie "fetchmail". Wzorując się na Torvaldsie, udostępnił swój program z obietnicą uaktualniania kodu źródłowego tak szybko i tak często, jak tylko będzie to możliwe. Gdy użytkownicy zaczęli nadsyłać raporty o błędach i sugestie zmian, spodziewał się jednego wielkiego bałaganu, ale ku swemu zaskoczeniu odkrył, że wszystko stanowiło zwartą całość. Analizując zwycięstwo podejścia Torvaldsa, doszedł do wniosku, że: używając Internetu jako "płytki Petriego⁴", a także narzędzia do dokonania surowej kontroli i naturalnej selekcji, Torvalds stworzył ewolucyjny model działania wolny od centralnego planowania.

Ponadto, jak sobie uświadomił Raymond, Torvalds znalazł sposób obejścia prawa Brooksa. Prawo to, po raz pierwszy sformułowane przez Freda P. Brooksa, menadżera projektu OS/360 w firmie IBM i autora wydanej w roku 1975 książki

The Mythical Man-Month⁵, gł osi ę że dodawanie kolejnych programistów⁶ do projektu powoduje jedynie dalsze jego opó żnienie. Raymond, jak wi ekszo ć hakerów wierzący dotychczas, ę zupa staje się tym gorsza, im wi ecej kucharzy, odkrył , ę Torvaldsowi udało się pokazać, ę jest przeciwnie. Zapraszając do kuchni coraz wi ecej i wi ecej kucharzy, czynił programistyczną zupe ę coraz lepszą⁷.

Raymond przelał swe spostrze enia na papier. Nadał im postać przemówienia pt. "The Cathedral and the Bazaar" (Katedra i bazar), które zapewne wygł osił dla grupy przyjaciół i sąsiadów z Chester County w Pensylwanii. Sposobowi zarządzania projektem GNU przeciwstawił tam styl zarządzania stosowany przez Torvaldsa i pozostał ych hakerów z grupy tworzącej jądro Linuksa. Jak twierdzi, spotkał osię to od razu z entuzjastycznym przyjęciem, choć nie tak entuzjastycznym, jak na zorganizowanym na wiosnę następnego roku (1997) kongresie niemieckich użytkowników Linuksa.

"Na tym kongresie po moim przemówieniu nastąpił aowacja na stojąco - wspomina Raymond. - Sądzę, ę wynikało to z dwóch przyczyn. Po pierwsze, ludzie byli podekscytowani tym, co usł yszełi, a po drugie, cieszyli się, ę zrozumieli pomimo bariery językowej".

Ostatecznie przemówienie zostało o publikowane pod tym samym tytuł em The Cathedral and the Bazaar. Tytuł był związany z gł ówną podkreślaną przez Raymonda analogią. Programy GNU to były "katedry", monumentalne, centralnie zaplanowane pomniki hakerskiej etyki, tworzone z myślą o wytrzymaniu próby czasu. Zaś Linux przypominał "wielki rozgadany bazar", był oprogramowaniem tworzonym przez lu żną, nieznającą centralizacji, dynamiczną naturę Internetu. Podtekstem tej analogii było porównanie Stallmana z Torvaldsem - Stallman był klasycznym architektem "katedr", wizjonerem, który potrafił zniknąć na 18 miesięcy, by potem pojawić się z czymś takim jak GNU C Compiler, zaś Torvalds przypominał raczej genialnego gospodarza przyjęcia. Pozwalając

innym wieść dyskusje i wkraczając jedynie, gdy wszyscy przy stole potrzebowali punktu odniesienia, stworzył model tworzenia oprogramowania będący odbiciem jego własnej osobowości skłonnej do pozostawania w cieniu wydarzeń i działania zza kulis. Z jego punktu widzenia nadrzędnym zadaniem menadżera nie jest narzucanie sterowania i kontroli lecz zapewnienie przepływu idei.

Podsumowując, Raymond stwierdził, że: "największym hakerskim osiągnięciem Linusa nie jest sama konstrukcja jądra Linuksa lecz stworzenie i konsekwentna realizacja nowego modelu tworzenia oprogramowania"⁸.

Doceniając menedżerski sukces Torvaldsa, Raymond odniósł własny sukces. Jednym ze słuchających go na niemieckim kongresie linuksowym był Tim O'Reilly z O'Reilly & Associates, firmy specjalizującej się w wydawaniu podręczników informatycznych i książek w różny sposób dotyczących programowania i oprogramowania (wydawcy również tej książki). Po wysłuchaniu przemówienia Raymonda O'Reilly od razu poprosił go o powtórzenie go na organizowanej przez firmę inauguracyjnej konferencji poświęconej językowi Perl, która miała się odbyć nieco później w tym samym roku w Monterey w Kalifornii.

Choć głównym tematem konferencji był Perl, język skryptowy stworzony przez uniksowego hakera Larry'ego Walla, O'Reilly zapewnił Raymonda, że inne zagadnienia dotyczące technologii wolnego oprogramowania również będą poruszane. Miał nadzieję, że coraz większe komercyjne zainteresowanie Linuksem i Apachem, popularnym wolnym serwerem WWW, pozwoli na rozpowszechnienie informacji o roli wolnego programowania w tworzeniu struktury Internetu. Od przyjaznych dla WWW języków w rodzaju Perla i Pythona, przez działające "zza kulis" takie programy jak BIND (ang. Berkeley Internet Naming Daemon), który pozwala użytkownikom stosować łatwiej zapamiętanie nazwy domen (np. amazon.com) zamiast tajemniczych numerów IP, po

najpopularniejszy program pocztowy sendmail - wolne oprogramowanie zaczynało stanowić coraz bardziej istotną część Internetu. Ale tym, którzy je tworzyli, brakowało samoświadomości, tak jak brakuje jej kolonii mrówek budujących piękny kopiec przez dokładanie pojedynczych ziarenek piasku. O'Reilly uznał, że przemówienie Raymonda nadaje się do jej rozbudzenia. Pokaże też, że ruch wolnego programowania nie zaczął się i nie kończy się na projekcie GNU. Właśnie języki Perl i Python oraz oprogramowanie internetowe BIND, Apache i sendmail były najlepszymi dowodami, że wolne oprogramowanie wszędzie znajduje swe miejsce, a jego znaczenie rośnie. Zapewnił także Raymonda, że spotka się z równie gorącym przyjęciem, jak w Niemczech, a może nawet gorętszym.

Miał rację. "Tym razem owacja na stojącą spotkała mnie już przed przemówieniem, a nie dopiero po nim" - mówi Raymond ze śmiechem.

Jak było do przewidzenia, audytorium składało się nie tylko z hakerów, lecz również z innych osób zainteresowanych rosnącą rolą ruchu wolnego oprogramowania. Jedna grupa reprezentowała Netscape z Mountain View w Kalifornii, firmę, której trzyletnia walka z Microsoftem o dominację na rynku przeglądarek WWW właśnie dobiegała końca.

Członkowie tej grupy, zaintrygowani przemówieniem Raymonda i dążący za wszelką cenę do odzyskania utraconych udziałów w rynku, przesłali komunikat do głównej siedziby firmy. Kilka miesięcy później, w styczniu 1998 Netscape ogłosił plan opublikowania kodu źródłowego swego flagowego produktu - przeglądarki WWW Navigator, licząc na wsparcie hakerów w dalszym rozwoju produktu.

Gdy dyrektor naczelny Netscape'a Jim Barksdale stwierdził, że artykuł Raymonda Cathedral and the Bazaar miał istotny wpływ na podjęcie tej decyzji, firma natychmiast uznała go za hakerską osobistość. Zdecydowany nie zmarnować nadarzającej się okazji, Raymond udał się na zachód, by

udzielać wywiadów, doradzać rządzącym Netscapem oraz wziąć udział w ostatniej fazie uroczystości udostępnienia kodu źródłowego Navigatora. Kod ten nosił nazwę "Mozilla", co było odniesieniem do jego gargantuicznych rozmiarów - 30 milionów linii oraz dziedzictwa. Mozilla była komercyjnym odrostem programu Mosaic, przeglądarki stworzonej przez Marca Andreessena z University of Illinois i doskonałym potwierdzeniem faktu, że większość programistów, budując nowy program, woli korzystać ze starych, dostępnych do modyfikacji kodów.

Będąc w Kalifornii, Raymond znalazł także czas na odwiedzenie VA Research, firmy z Santa Clara sprzedającej stacje robocze z preinstalowanym systemem GNU/Linux. Zdecydował się tam na spotkanie z niewieloma osobami: z założycielem firmy Larrym Augustinem, kilkoma pracownikami oraz z Christine Peterson, prezesem zarządu Foresight Institute¹⁰, "think tanku¹¹" z Doliny Krzemowej zajmującego się nanotechnologią.

Plan spotkania został ograniczony do jednego zagadnienia: w jaki sposób sprawić, aby inne firmy podążyły tą samą drogą co Netscape? Raymond nie pamięta rozmowy, ale zapamiętał pierwsze zgłoszone zażalenie. Pomimo wysiłków Stallmana i innych hakerów, by uświadomić ludziom, że słowo "wolny" w określeniu "wolne oprogramowanie" oznacza uwolnienie od ograniczeń, a nie od kosztów, świadomość tego faktu jeszcze nie stała się powszechna. Większość zarządzających instytucjami, słysząc ten termin po raz pierwszy, uważa słowo "wolny" za synonim "zerowych kosztów¹²" i odrzuca wszelkie możliwości innej interpretacji. Dopóki hakerzy nie rozwiążą problemu tej podwójnej interpretacji, dopóty ruch nie osiągnie sukcesów, nawet po decyzji Netscape' a

Peterson, której organizacja była żywo zainteresowana dalszym rozwojem wolnego oprogramowania, zaproponował a termin alternatywny: oprogramowanie open source.

Patrząc wstecz, Peterson twierdzi, że termin "oprogramowanie

open source" wymyślił a podczas omawiania decyzji Netscape'a z przyjacielem z branży public relations¹³. Nie pamięta gdzie na ten pomysł wpadła, a także czy go skądś zapożyczył a, ale pamięta, że przyjacielowi się on nie podobał¹⁴.

Na spotkaniu reakcja była diametralnie różna. "Wahałem się, czy tę nazwę sugerować - wspomina Peterson. - Nie miałem żadnych stałych kontaktów z tą grupą, więc użyłam terminu przypadkowo, bez podkreślania, że jest nowy." Ku jej zaskoczeniu, nowe określenie chwyciło. Pod koniec spotkania większość uczestników, łącznie z Raymondem, zdawała się używać go z przyjemnością.

Raymond twierdzi, że publicznie terminu "oprogramowanie open source" zamiast "wolne oprogramowanie" po raz pierwszy użył dopiero w dzień lub dwa po rozmowie na lunchu w sprawie Mozilli, podczas której O'Reilly zaproponował zorganizowanie "spotkania na szczycie" w sprawie wolnego oprogramowania (Freeware Summit). O'Reilly chciał zwrócić uwagę mediów i społeczeństwa również na inne realizowane projekty, które także wpłynęły na decyzję Netscape'a o uwolnieniu Mozilli.

"Wszyscy ci faceci mieli tak wiele wspólnego, iż trudno było pojąć, że nic o sobie nawzajem nie wiedzą - twierdził O'Reilly. - Chciałem, aby świat się dowiedział, jak wielki był dotychczasowy wpływ kultury wolnego oprogramowania. Ludzie wiele tracili, nie znając tej tradycji".

Układając listę zaproszonych, O'Reilly podjął decyzję mającą dalekosiężne konsekwencje polityczne. Zdecydował się ograniczyć zaproszenia jedynie do twórców oprogramowania z zachodniego wybrzeża, takich jak Wall, Eric Allman - twórca sendmaila i Paul Vixie - twórca BIND-a. Były oczywiście wyjątki, jak pochodzący z Pensylwanii Raymond przebywający akurat w mieście z powodu spotkania w sprawie Mozilli. Został także zaproszony mieszkający w Wirginii Guido van Rossum, twórca Phytona. "Zaprosił go bez uzgadniania ze mną Frank Willison, mój redaktor naczelny i najlepszy wśród nas znawca Pythona - wspomina O'Reilly. - Byłem zadowolony z jego

przyjazdu, choć, wedle moich zamierzeń, miało to być lokalne spotkanie".

Niektórzy uznali, że niezaproszenie Stallmana było afrontem i zadzieraniem nosa. "Z tego powodu zdecydowałem, by nie brać w tym udziału" - mówi Perens, wspominając "szczyt". Raymond, który przyszedł, wspomina, że argumentował za zaproszeniem Stallmana, ale bez skutku. Oskarżenie o afront zostało wzmocnione faktem, że między gospodarzem spotkania O'Reillym i Stallmanem istniała azadawniona i powszechnie znana wrogość z powodu sporu o prawa autorskie instrukcji do oprogramowania. Przed spotkaniem Stallman ogłosił, że instrukcje do wolnego oprogramowania można dowolnie kopiować i modyfikować tak samo jak samo wolne oprogramowanie. Tymczasem O'Reilly argumentował, że komercyjne podręczniki są wartością dodaną, zwiększającą użyteczność wolnego oprogramowania, dzięki czemu staje się ono dostępne dla szerszych kręgów społeczeństwa.

Przedmiotem sporu był również tytuł spotkania - Stallman upierał się przy "Free Software", zamiast mającego mniejszą wagę polityczną "Freeware"¹⁵.

Patrząc wstecz, O'Reilly nie uważa, że niezaproszenie Stallmana był oafrotem. "Do tego czasu nigdy nie spotkałem się osobiście ze Stallmanem, ale wymiana e-maili między nami świadczyła, że jest on osobą nieskłoną do jakichkolwiek kompromisów i niechętną do podejmowania dyskusji. Chciałem, aby na spotkaniu była reprezentowana tradycja GNU i dlatego zaprosiłem Johna Gilmore'a i Michala Tiemanna, których znałem osobiście i którzy byli szalenie przywiązani do GPL, ale chętniej dyskutowali o zaletach i słabościach projektów wolnego oprogramowania i jego tradycji. Mając na uwadze rwetes, który spowodowało pominięcie Richarda, sądzę, że powinienem go zaprosić, ale nadal nie uważam, że mój błąd można interpretować jako demonstrację braku szacunku dla projektu GNU i samego Richarda".

Z afrontem lub bez, O' Reilly i Raymond twierdzą, że termin

"open source" został przyjęty bez oporów przez tak znaczną większość uczestników spotkania, że można to uważać za sukces. Podzielili się oni pomysłami i doświadczeniami oraz uczestniczyli w burzy mózgów, której celem było znalezienie najlepszych sposobów poprawienia wizerunku wolnego oprogramowania. Zastanawiano się głównie, jak wykorzystać sukces wolnego oprogramowania, zwłaszcza w świecie Internetu, w wielkiej rozgrywce GNU/Linuksa z Microsoft Windows. Ale tak jak podczas wcześniejszego spotkania w VA dyskusja szybko zbiegła na problemy związane z samym terminem "wolne oprogramowanie". Gospodarz spotkania, O'Reilly, pamięta szczególnie refleksyjny komentarz biorącego udział w spotkaniu Torvaldsa:

"Linus, który niedawno przeniósł się do Krzemowej Doliny, wyjaśnił, że dopiero ostatnio dowiedział się, że po angielsku słowo 'free' ma dwa znaczenia: 'wolny' i 'bezpłatny'".

Michael Tiemann, założyciel Cygnusa, zaproponował rozwiązanie problemu przez zastąpienie terminu "free software" (wolne oprogramowanie) przez "sourceware" (mniej więcej oprogramowanie ze źródłem), ale jak wspomina O'Reilly, "nikomu się to zbytnio nie spodobało. Wówczas Eric zaproponował 'Open source'".

Choć niektórym termin bardzo się spodobał, niewielu poparło zmianę oficjalnej terminologii. Na koniec jednodniowej konferencji poddano pod głosowanie trzy nazwy: "free software", "open source" i "sourceware". O'Reilly twierdzi, że termin "open source" zwyciężył 9 głosami na 15 głosujących. Choć niektórzy zgłaszali obiekcje, ostatecznie zgodzono się na używanie wybranej nazwy przy późniejszych kontaktach z prasą.

"Chcieliśmy zmanifestować solidarność" - mówi O'Reilly.

Wkrótce nowe określenie znalazło swe miejsce w narodowym słowniku. Zaraz po zakończeniu spotkania O'Reilly zaprosił jego uczestników na konferencję prasową, na której zjawili się reporterzy z "New York Timesa", "Wall Street Journal" i innych liczących się pism. W ciągu kilku miesięcy twarz Torvaldsa

znalazł się na okładce magazynu Forbes, w którego wnętrzu umieszczono zdjęcia Stallmana, Larry'ego Walla - twórcy Perla i Briana Behlendorfa - lidera grupy, która stworzyła Apache. Oprogramowanie "open source" stało się rzeczywiście otwarte dla biznesu.

Dla biorących udział w spotkaniu, np. dla Tiemanna, najważniejsza była manifestacja solidarności. Choć jego firma odniosła znaczny sukces, sprzedając narzędzia wolnego oprogramowania i związane z nimi usługi, rozumiał trudności piętrzące się przed innymi programistami i przedsiębiorcami. "Chyba nikt nie zaprzeczy, że używanie słowa 'free' w wielu sytuacjach powoduje zamieszanie - mówi Tiemann. - 'Open source' robi wrażenie terminu bardziej przyjaznego i sensownego dla biznesu, zaś 'wolne oprogramowanie' było terminem, za którym stały racje moralne. Uznaliśmy, że lepiej na dobre i na złe przyjąć się do tego umu zwolenników oprogramowania open source."

Stallman nie śpieszył się z wyrażeniem opinii o nowym terminie "open source". Najpierw krótko zastanawiał się nad przyjęciem nowej nazwy, lecz szybko ją odrzucił. "Wiem o tym, gdyż z nim na ten temat rozmawiałem" - twierdzi Raymond.

Przed końcem roku 1998 Stallman sformułował swoje stanowisko: "open source" jest nazwą pomocną przy informowaniu o technicznych korzyściach stosowania wolnego oprogramowania, lecz ośmielał mówców do łagodzenia swego stanowiska w kwestiach dotyczących wolnego oprogramowania. Z powodu tej wady Stallman postanowił pozostać wierny starej nazwie wolnego oprogramowania.

Na LinuxWorld Convention and Expo w roku 1999 podsumował swoje stanowisko i mimo iż nazwany został przez Torvaldsa "przemijającą częścią" linuksowej społeczności, błagał kolegów hakerów o nieuleganie powabom łatwego kompromisu.

"Ponieważ pokazaliśmy już, na jak wiele nas stać, nie musimy desperacko zabiegać o współpracę firm i kompromisowo rezygnować z wyznaczonych celów - mówił podczas dyskusji

panelowej. - Pozwólmy im na złożenie oferty, którą moglibyśmy zaakceptować. Nie musimy nic zmieniać w naszym postępowaniu, aby pozyskać ich pomoc. Dążąc we właściwą stronę, możecie najpierw wykonać jeden krok, potem następny, potem jeszcze jeden i jeszcze jeden, aż wreszcie dotrzecie do celu. Lub możecie działać połowicznie i przystać na kompromis, lecz wówczas nigdy nie wykonacie drugiego kroku i nigdy nie dotrzecie do celu."

Ale już przed konferencją LinuxWorld dążenie Stallmana do zdystansowania się od swych bardziej ugodowych towarzyszy stawało się coraz bardziej widoczne. W kilka miesięcy po Freeware Summit O'Reilly zorganizował drugą doroczną konferencję języka Perl (Perl Conference). Tym razem Stallman był obecny. Podczas dyskusji panelowej, na której IBM ogłosił decyzję o umieszczeniu w swej ofercie komercyjnej wolnego serwera WWW Apache, zawł adnął mikrofonem, aby przed zebraną publicznością wygłosić tyradę przeciwko Johnowi Ousterhoutowi, twórcy języka skryptowego Tcl. Nazwał Ousterhouta pasożytem żerującym na społeczności wolnego oprogramowania, a to z powodu sprzedawania przez jego firmę Scriptics komercyjnej wersji Tcl. "Nie uważam, aby do dalszego istnienia Tcl potrzebne był oistnienie Scriptics" - gł ośł Stallman pomimo syków zgromadzonych słuchaczy¹⁶.

"Był a to obrzydliwa scena - wspomina Rich Morin z Prime Time Freeware. - John zrobił rzeczy, które należy docenić: Tcl, Tk i Sprite¹⁷. To jego znaczny dorobek".

Pomimo sympatii dla Stallmana i jego osiągnięć wyczuwa się u Morina empatię wobec sł uchaczy, którzy z powodu jego niestosownego zachowania znaleźli się w krępującej sytuacji. Wybuch Stallmana na Perl Conference odstraszył innego potencjalnego sympatyka, Bruce'a Perensa. W roku 1998 Eric Raymond zaproponował stworzenie Open Source Initiative czyli OSI, organizacji, która miał a strzec porządku stosowania nazwy "open source" i określać zakres stosowania dla firm zainteresowanych rozwijaniem własnych programów. Raymond

poprosił Perensa o sformułowanie definicji¹⁸.

Perens odszedł potem z OSI, wyrażając żal, że organizacja stanęła w opozycji do Stallmana i Fundacji Wolnego Oprogramowania (FSF). Patrząc wstecz, uznaje ówczesną potrzebę zdefiniowania wolnego oprogramowania poza auspicjami FSF i rozumie, dlaczego część hakerów nadal woli trzymać się od niej z daleka. "Rzeczywiście, lubię i szanuję Richarda - twierdzi - ale sądzę, że wykonał by swoje zadanie lepiej, gdyby dokładniej wyważał racje. Mógłby przez kilka miesięcy odpocząć od wolnego oprogramowania".

Jego monomaniakalna energia nie była w stanie przeciwstawić się oddziaływaniu na opinię publiczną grupy z rozmachem proponującej oprogramowanie open source. Gdy w sierpniu 1998 producent procesorów Intel wykupił część akcji firmy Red Hat, dostawcy GNU/Linuksa, "New York Times" w omawiającym ten zakup artykule napisał, że firma ta jest produktem ruchu "nazywanego alternatywnie ruchem wolnego oprogramowania lub oprogramowania open source"¹⁹. Sześć miesięcy później John Markoff w artykule na temat Apple Computer poinformował o "adoptowaniu" przez tę firmę serwera Apache należącego do oprogramowania open source, umieszczając to określenie w tytule²⁰.

Zbiegło się to z dążeniem wielu firm, które zaczęły używać terminu "open source". Jeszcze przed sierpniem 1999 Red Hat i inne firmy zajmujące się oprogramowaniem open source i stosujące tę nazwę zaczęły sprzedawać swe akcje na giełdzie Nasdaq²¹. W grudniu VA Linux, poprzednio VA Research, osiągnęło historyczne przebicie ceny IPO²². Cena otwarcia wynosząca 30 dolarów za akcję podskoczyła gwałtownie aż do 300 dolarów, by przed zamknięciem spaść jedynie do 239 dolarów. Szczęśliwcy, którym udało się kupić akcje po cenie zbliżonej do dolnego progu, mogli osiągnąć 698% zysku, co było rekordem giełdy Nasdaq.

Wśród nich był Eric Raymond, który jako członek zarządu od czasów uruchomienia Mozilli przy debiucie giełdowym VA Linux

otrzymał 150 tys. akcji. Zdumiał się faktem, że opisanie odmienności stylów menedżerskich Stallmana i Torvaldsa przyniosło mu potencjalny majątek wart 36 milionów dolarów i zasiadł do napisania kolejnego artykułu. Rozwodził w nim wzajemne zależności etyki hakerskiej i bogactwa:

Ostatnio dziennikarze często pytają, czy sądzę, że społeczność open source zostanie skorumpowana napływem wielkich pieniędzy. Komercyjne zapotrzebowanie na programistów od dawna jest tak duże, że wszyscy, których zarabianie pieniędzy bardzo pociągało, już odeszli. W naszej społeczności został już dokonana autoselekcja i pozostali ci, dla których najważniejsze był coś innego: duma, artystyczna pasja i inne podobne rzeczy²³.

Nie wiadomo, w jakim stopniu podobne komentarze uśmierzyły y podejrzenia, że Raymond i inni twórcy ruchu open source działali po prostu dla pieniędzy, ale niewątpliwie uświadomili hakerskiej społeczności, że wszystko, czego potrzeba, aby sprzedać koncepcję wolnego oprogramowania, to życzliwe nastawienie i trafiająca do adresatów informacja. Zamiast za wszelką cenę zwalczać rynek, jak robił to Stallman, Raymond i Torvalds oraz inni, nowi przywódcy hakerskiej społeczności przyjęli o wiele luźniejsze zasady działania, ignorując pewne obszary rynku, a oddziałując na inne. Zamiast grać rolę tkwiących na uczelniach wygnańców, wykorzystywali zdobyte poważanie do powiększania swych możliwości.

"W swoje najgorsze dni Richard wierzy, że Linus Torvalds i ja zmówiliśmy się, by przejąć jego rewolucję - mówi Raymond. - Odrzucenie przez niego terminu 'open source' i jego rozmyślne, uparte tworzenie ideologicznego rozłamu bierze się, jak sądzę, z mieszaniny idealizmu i chęci zawłaszczania terytorium. Wielu ludzi sądzi, że wynika to z jego ego. Nie wierzę w to. Po prostu idea wolności oprogramowania tak bardzo stopiła się z jego osobowością, że wszelkie jej zagrożenia odczuwa jako groźbę wobec siebie samego".

Ironią losu może się wydawać fakt, że ani open source ani jego

zwolennicy nie zmniejszyli przywódczej roli Stallmana. Ciągła za nim następni kandydaci do nawrócenia. Jednakże rzucone przez Raymonda oskarżenie o dążenie do panowania na własnym terytorium, nic nie straciło o swojej przekłębnej aktualności. Jest wiele przykładów, gdy chwytła on za broń bardziej ze zwyczaju niż potrzeby. Takim działaniem było na przykład wstępne odrzucenie jądra Linuksa oraz jego trwająca wciąż niechęć do działań politycznych poza światem oprogramowania.

Ostatnie debaty na temat oprogramowania open source również pokazują, iż Stallman pozostaje przywiązany do metod, które dotychczas pozwalały mu na działanie. "Jedną z podstawowych cech jego charakteru jest niewzruszoność - mówi Ian Murdock. - Może przez dziesięciolecie tkwić w jednym miejscu, czekając, aż wokół zbiorą się ludzie, którzy podzielą jego poglądy". Murdock uważa tę niewzruszoność natury Stallmana za wartościową i inspirującą. Stallman może już nie być jedynym przywódcą ruchu wolnego oprogramowania, ale pozostanie dla tej społeczności gwiazdą świecącą nad biegunem. "Możecie być pewni, że zachowa wierność swym poglądom - mówi Murdock, - czego większość ludzi u niego nie lubi, ale nawet nie zgadzając się z nim, musi przyznać, że go szanuje".

Przypisy

1 O ile terminy "free software" i "Free Software Foundation (FSF)" mają powszechnie przyjęte polskie odpowiedniki - "wolne oprogramowanie" i "Fundacja Wolnego Oprogramowania", o tyle "open source" i odpowiednio "Open Source Initiative - OSI" są tłumaczone różnie. Mamy więc najbardziej dosłowne "otwarte oprogramowanie", a także "otwarty kod źródłowy" i "ogólnodostępny kod źródłowy", jak również wersję coraz powszechniej używaną - "oprogramowanie open source". Ta wersja jest stosowana np. na stronie WWW polskiej sekcji Fundacji Wolnego Oprogramowania (patrz:

<http://www.gnu.org/philosophy/free-software-for-freedom.pl.html>) oraz na tworzonej właśnie stronie OpenPoland.org, której robocza wersja jest dostępna pod adresem http://openpoland.org/sth_index.php - przyp. tł um.

2 Patrz: Peter Salus, FYI-Conference on Freely Redistributable Software, 2/2, Cambridge (1995) (zarchiwizował Terry Winograd), <http://hci.stanford.edu/pcd-archives/pcd-fyi/1995/0078.html>

3 Choć Linus Torvalds jest Finem, jego językiem ojczystym jest szwedzki. W The Rampantly Unofficial Linus FAQ znajdujemy krótkie wyjaśnienie:

W Finlandii żyje znacząca (około 6%) mniejszość mówiąca po szwedzku. Sami siebie nazywają "finlandssvensk" lub "finlandssvenskar" i uważają się za Finów. Często ich rodziny żyją w Finlandii od wieków. Szwedzki jest jednym z dwóch oficjalnych języków Finlandii.

<http://tuxedo.org/~esr/faqs/linus/>

4 Specjalne naczynie z luźną pokrywką, zrobione ze szkła lub z tworzywa sztucznego, służące do rozmnażania mikroorganizmów. Nazwa pochodzi od bakteriologa niemieckiego Juliusa Richarda Petriego (1852-1921) - przyp. tł um.

5 Można to przełożyć jako "Mityczny osobomiesiąc". W Internecie znalazł emrównież tł umaczenie "Mityczny osobodzień". Widocznie nasza biurokracja osobodni przedkłada nad osobomiesiące - przyp. tł um.

6 "Adding manpower to a late software project makes it later" czyli "Dodanie kolejnych pracowników do opóźnionego projektu programistycznego powoduje jego jeszcze większe opóźnienie" - przyp. tł um.

7 Prawo Broksa jest krótkim, poręcznym podsumowaniem następującego fragmentu książki Brooksa:

"Ponieważ tworzenie oprogramowania jest z natury rzeczy

działaniem systemowym, doświadczaniem wzajemnych, złożonych relacji, wysiłek stąd wynikający pochłaniania masy energii i niweczy efekty zmniejszenia zadań powierzanych pojedynczym osobom. W rezultacie dodawanie większej liczby osób wydłuża czas wykonania zadania, miast go skracać. Patrz: Fred P. Brooks, The Mythical Man-Month (Addison Wesley Publishing, 1995).

8 Patrz: Eric Raymond, The Cathedral and the Bazaar (1997).

9 BIND jest tzw. serwerem nazw domen czyli narzędziem tłumaczącym nazwy domen na numery IP (Internet Protocol - protokołu internetowego)
- przyp. tłum.

10 Foresight to przezorność, zapobiegliwość, patrzenie w przyszłość. Foresight Institute to instytut zajmujący się technologiami przyszłości - przyp. tłum.

11 Według internetowego słownika wyrazów obcych Wł. Kopalińskiego (<http://www.slownik-online.pl/index.php>): think tank - grupa a. ośrodek powołane przez rząd a. przemysł dla intensywnych studiów teoretycznych, zwł. problemów kultury i społeczeństwa, oraz badań naukowo-technicznych. Etym. - slang amer.ang. z think "myśleć" i tank "zbiornik" - przyp. tłum.

12 Przypominam, że po angielsku słowo "free" oznacza nie tylko "wolny", lecz także "bezpłatny", co utrudnia rozeznanie - przyp. tłum.

13 Według internetowego słownika wyrazów obcych Wł. Kopalińskiego (<http://www.slownik-online.pl/index.php>): public relations - amer. ang., całokształt stosunków (opinii, postaw, poglądów) społ. do działalności danego przedsiębiorstwa, org., instytucji; działalność osoby, firmy, instytucji w celu osiągnięcia dobrych stosunków z innymi osobami, firmami itd. a. zdobycia życzliwości opinii publ.; metody stosowane dla osiągnięcia tych celów; personel, zespół trudniący się takimi działaniami zawodowo - przyp. tłum

14 Patrz: Malcolm Maclachlan, Profit Motive Splits Open Source Movement, TechWeb News (26 sierpnia 1998).

15 Są to niuanse językowe związane z dwuznacznością słowa "free" jako "wolny" i "bezpłatny". Stallman zawsze podkreślał, że "free software" oznacza oprogramowanie wolne, a nie bezpłatne, tymczasem słownik The American Heritage(r) Dictionary of the English Language: Fourth Edition. 2000. tłumaczy "freeware - oprogramowanie dostępne bezpłatnie, zwykle przez Internet" - przyp. tłum.

16 Patrz: Malcolm Maclachlan, Profit Motive Splits Open Source Movement, TechWeb News (26 sierpnia 1998).

<http://content.techweb.com/wire/story/TWB19980824S0012>

17 Tcl (ang. Tool Command Language - język poleceń narzędziowych) - język skryptowy, Tk (ToolKit) - rozszerzenie Tcl, zestaw narzędzi graficznych służących do budowy interfejsów graficznych. Podstawowe informacje można znaleźć w wydany przez Helion Leksykonie kieszonkowym Tcl/Tk - patrz: <http://helion.pl/ksiazki/tcltk.htm>

Sprite - Sprite is a research operating system developed at the University of California, Berkeley, by John Ousterhout's research group. Badawczy system operacyjny stworzony na University of California w Berkeley przez grupę kierowaną przez Johna Ousterhouta - więcej informacji można znaleźć np. na stronie

<http://www.cs.berkeley.edu/projects/sprite/sprite.html> - przyp. tłum.

18 Patrz: Bruce Perens et al., The Open Source Definition, The Open Source Initiative (1998).

<http://www.opensource.org/docs/definition.html>

19 Patrz: Amy Harmon, For Sale: Free Operating System, New York Times (28 września 1998).

<http://www.nytimes.com/library/tech/98/09/biztech/articles/28linux.html>

20 Patrz: Markoff, Apple Adopts 'Open Source' for its Server Computers, New York Times (17 marca 1999).

<http://www.nytimes.com/library/tech/99/03/biztech/articles/17apple.html>

21 NASDAQ (ang. National Association of Securities Dealers Automated Quotation System - automatyczny system notowań Narodowego Związku Maklerów Papierów Wartościowych), największa na świecie elektroniczna giełda papierów wartościowych - przyp. tłum.

22 Gdy korporacje sprzedają akcje, oddają część kontroli nad firmą w ręce akcjonariuszy (których głównie interesują zyski i dywidendy), a w zamian otrzymują gotówkę, aby móc rozwijać swój biznes. Pierwsza sprzedaż akcji dla pierwotnych inwestorów, którymi są głównie banki inwestycyjne (ang. investment banks) nosi nazwę Initial Public Offering (pierwsza publiczna oferta sprzedaży), w skrócie IPO. Akcje zakupione przez banki inwestycyjne są sprzedawane na rynku wtórnym (ang. secondary market). Hot IPOs często osiągają na rynku wtórnym wielokrotnie wyższe ceny, ale niestety, dostęp do IPOs po cenie pierwotnej jest bardzo ograniczony.

Wyjaśnienie pochodzi z poradnika inwestorów giełdowych - <http://www.polishinvestor.com/Polski/akcje.html> - przyp. tłum.

23 Patrz: Eric Raymond, Surprised by Wealth, Linux Today (10 grudnia 1999).

http://linuxtoday.com/news_story.php3?ltsn=1999-12-10-001-05-NW-LF

12. KRÓTKA PODRÓŻ PRZEZ HAKERSKIE PIEKŁO

Richard Stallman bez zmrużenia powiek spoglądał przez przednią szybę wynajętego samochodu, oczekując na zmianę świateł, by ruszyć w dalszą drogę przez centrum Kihei.

Jechaliśmy do pobliskiego miasta Pa'ā, gdzie mniej więcej przed godziną powinniśmy się byli spotkać na obiedzie z kilkoma programistami i ich żonami.

Znaleźliśmy się tu w dwie godziny po wykładzie Stallmana w

Maui High Performance Center, a Kihei, które przed wykładem zdawało się być miastem przyjaznym dla przybyszów, teraz sprawiało kłopoty na każdym kroku. Jak wszystkie kurorty nadmorskie zdawał o się być przedmieściem rozciągniętym nad miarę w jednym wymiarze. Jadąc wzdłuż głównej ulicy wypełnionej niekończącymi się straganami z hamburgerami, agencjami nieruchomości i sklepami z bikini, miałem wrażenie, że jesteśmy drobinami uwięzionymi w blaszanym pancerzu samochodu i przesuwającymi się wzdłuż przewodu pokarmowego gigantycznego handlowego potwora. Wrażenie to potęgował brak dodatkowych pasów dla ruchu lokalnego, co powodowało, że na jedynej zatłoczonej jezdni posuwanie się do przodu przypominało o konwulsje drgającej sprężyny. Zielone światła zapalające się 200 metrów przed nami, gdy do nich dojeżdżaliśmy, zmieniały się na żółte.

Stallmana, który całe życie mieszkał na wschodnim wybrzeżu, o zawał mógł a przyprawić perspektywa spędzenia większej części słonecznego hawajskiego popołudnia w pułapce ruchu ulicznego. Jeszcze gorsza była świadomość, że skrócenie w prawo mniej więcej ćwierć mili wcześniej pozwoliłoby na ominięcie tej pułapki. Niestety, byliśmy na łasce jadącego przed nami programisty z laboratorium, który znał trasę i który zdecydował się przewieźć nas malowniczą trasą przez Pa'ia zamiast sąsiednią autostradą Pilani.

"To straszne - mówił Stallman między kolejnymi zmianami światła. - Dlaczego nie pojechaliśmy inną drogą?"

Zielone światła ponownie zapaliły się ćwierć mili przed nami.

Znowu przepeliliśmy do przodu kilka długości samochodu.

Powtarzało się to przez kolejne 10 minut, aż wreszcie dotarliśmy do głównego skrzyżowania umożliwiającego skręt na sąsiednią autostradę.

Kierowca przed nami zignorował to i przejechał przez skrzyżowanie na wprost.

"Dlaczego on nie skręca? - narzekał Stallman, wznosząc ręce w rozpacz. - Możesz to pojąć?"

Postanowiłem w ogóle nie odpowiadać. Fakt, że w Maui jadę samochodem ze Stallmanem jako kierowcą i tak wydawał się trudny do uwierzenia. Jeszcze dwie godziny temu nie miałem nawet pojęcia, czy Stallman w ogóle potrafi prowadzić samochód. Teraz, słuchając przez samochodowe stereo Yo-Yo Ma i grającego na wiolonczeli rzewne basowe nuty "Appalachian Journey" i oglądając po lewej zachód słońca, uznałem, że najlepiej zapaść się w miękką tapicerkę siedzenia.

Kiedy kolejna możliwość skręcenia w bok została azmarnowana, Stallman włączył prawy kierunkowskaz, by w ten sposób dać wskazówkę jadącemu przed nami. Nic z tego. Przepeliliśmy na wprost przez kolejne skrzyżowanie, utykając w ruchu na 200 metrów przed kolejnymi światłami. Stallman zsiniał z wściekłości.

"Tak jakby nas specjalnie ignorował" - powiedział i w daremnym wysiłku przyciągnięcia uwagi jadącego przed nami począł wykonywać gesty przypominające znaki dawane na lotnisku przez kierującego ruchem lądujących samolotów. Nasz przewodnik był jednak niewzruszony i przez następne pięć minut mogliśmy jedynie oglądać fragment głowy widoczny w lusterku wstecznym jego samochodu.

Spojrzałem przez okno po stronie Stallmana. Pobliskie wyspy Kahoolawe i Lanai pięknie wyglądały w oprawie zachodzącego słońca. Widok zapierał dech w piersi i jego wspaniałość mogła czynić znośniejszym życie tubylców. Starałem się zwrócić na to uwagę Stallmana, ale wściekły, jedynie mnie ofuknął.

Gdy nasz przewodnik przejechał przez kolejne skrzyżowanie, zupełnie ignorując napis "Autostrada Pilani na prawo", zacisnąłem zęby. Przypomniałem sobie, jak programista BSD Keith Bostic ostrzegał mnie, że "Stallmanowi niełatwo ścierpieć głupców". Twierdził, że każdemu, kto mówi lub robi coś głupiego, on mówi wprost: "To głupie".

Patrząc na nieświadomego sytuacji kierowcę przed nami, pojąłem, że to jego głupota, a nie niedogodność podróży doprowadzają Stallmana do szału.

"Wygląda na to, że wybierając tę drogę, w ogóle nie zastanowił się nad efektywnym dotarciem do celu" - powiedział Stallman. Słowo "efektywny" zawisło nad nami swym niewidocznym ciężarem. Niewiele jest rzeczy irytujących hakera bardziej niż nieefektywność. To właśnie nieefektywność wielokrotnego sprawdzania, czy drukarka się nie zacięła, spowodowała, że Stallman zaczął szukać dostępu do kodu źródłowego jej oprogramowania. To właśnie nieefektywność kodów zawłaszczonych przez komercyjnych dostawców oprogramowania doprowadziła do jego walki z firmą Symbolics i stworzenia projektu GNU. Jeżeli Jean Paul Sartre stwierdził kiedyś, że piekłem są inni ludzie, to męki piekielne hakera są spotęgowane przez ludzką głupotę, a całe życie Stallmana było dążeniem, by uchronić ludzkość przed stoczeniem się w te piekielne otchłanie.

W miarę naszego powolnego brnięcia do przodu ta piekielna metafora stawała się coraz bardziej oczywista. Mnogość sklepów, parkingów i z rzadka zapalane lampy uliczne Kihei przywołały na myśl nie miasto, lecz raczej marnie zaprojektowany, rozwlekły program komputerowy. Zamiast skierować ruch pojazdów na sąsiednie ulice i drogi ekspresowe, planiści miejscy zepchnęli wszystko do jednego zatłoczonego cieku głównej drogi. Dla hakera siedzenie w samochodzie w środku tego bałaganu było jak słuchanie nagranego na płcie CD zgrzytu paznokci na szkolnej tablicy.

"Niedoskonałość systemów doprowadza hakerów do szału - zauważył Steven Levy, co było kolejnym ostrzeżeniem, jakiego mi udzielono przed wejściem do samochodu prowadzonego przez Stallmana. - Właśnie dlatego hakerzy nie lubią prowadzić samochodów w ruchu, w którym zapalające się przypadkowo czerwone światła dziwnie ułożone ulice jednokierunkowe powodują niepotrzebne opóźnienia. Wywołuje to u nich chęć zmiany ustawienia znaków, przeprogramowania skrzynek sterujących światłami... innego zaprojektowania całego systemu"2.

Ale najbardziej denerwująca był a głupota przewodnika, któremu zawierzyliśmy. Zamiast znaleźć skrót umożliwiający jak najszybsze dotarcie do celu - co powinien uczynić instynktownie każdy haker - właczył nas do gry wymyślonej przez miejskich planistów. Jak Wergiliusz w "Piekło" Dantego postanowił odbyć z nami pełny objazd tego hakerskiego piekła, niezależnie od tego, czy mieliśmy na to ochotę.

Zanim zdążył em podzielić się ze Stallmanem tą opinią, nasz przewodnik wreszcie zapalił prawy kierunkowskaz. Stallman odprężył się nieco i napięcie w samochodzie na chwilę zelżało. Wróciło, gdy kierowca przed nami zwolnił. Po obu stronach ulicy widać było znaki "Roboty drogowe", a pasma drogi dojazdowej do odległej o ćwierć mili autostrady Pilani blokował beczynnie stojący buldożer i dwie sterty gruzu.

Dopiero po chwili Stallman zdołał zorientować się, że nasz przewodnik manewruje, by zawrócić. Gdy jego wzrok padł na buldożer i napis "Przejazd zamknięty", wszystko się w nim zagotowało.

"Dlaczego, dlaczego, dlaczego? - zaskowyczał. - Powinieneś wiedzieć, że tu nie ma przejazdu. Powinieneś wiedzieć, że tu nie przejedziesz. Robisz to specjalnie".

Kierowca zakończył manewr i minął nas, wracając na główną ulicę. Potrząsnął przy tym głową i usprawiedliwiająco wzruszył ramionami. Ten gest w połączeniu z zębami wyszczerzonymi w uśmiechu był podkreśleniem kontrastu między frustracją przybysza z kontynentu a spokojnym fatalizmem wyspiarza. Przez zamknięte szyby wynajętego przez nas samochodu dotarł do nas sens tego zwięzłego komunikatu "To jest Maui i co począć?"

Tego już Stallman nie mógł znieść.

"Mam dość twego pieprzonego uśmiechu - wrzasnął, przyćmiewając mgiełką oddechu zamkniętą szybę. - To twoja pieprzona wina. Wszystko poszłoby łatwiej, gdybyś to robił moim sposobem".

Stallman słowa "moim sposobem" podkreślił dwukrotnym

uchwyceniem i pociągnięciem do siebie kierownicy.

Przypominało to histerię dziecka, które siedząc za kierownicą, na niej wyładowuje swą złość. Dźwięki, jakie z siebie wydawał, wskazywały, że jest wściekły, udreńczony i bliski łez.

Na szczęście łez się nie pojawił. Jego furia ustała tak nagle, jak nagle się pojawił. Po kilku jęśliwych sapnięciach zawrócił samochód. Gdy ponownie znaleźliśmy się na głównej ulicy, jego twarz była już tak spokojna jak 30 minut temu, gdy wyjeżdżaliśmy z hotelu.

Po pięciu minutach dotarliśmy do następnego skrzyżowania.

Tym razem ławo wjechaliśmy na autostradę i po kilku sekundach mknęliśmy w stronę Pa' a, napawając się uspokajającą szybkością. Słońce jeszcze niedawno jasno żółtało się po lewej stronie Stallmana, teraz świeciło w lusterku wstecznym pomarańczową czerwienią, tak jakby pożyczyło barwy od drzew wili wili obrastających obie strony autostrady. Przez następnych dwadzieścia minut słyszeliśmy jedynie dźwięk silnika samochodowego, szum opon na asfalcie oraz dźwięki wiolonczeli i skrzypiec brzmiących smętnymi ludowymi nutami pieśni z Appalachów.

Przypisy

1 Z biografią i osiągnięciami Yo Yo Ma można się zapoznać na stronie <http://www.yo-yoma.com/> - przyp. tłum

2 Patrz: Steven Levy, Hackers (Penguin USA [okładka miękka], 1984), s. 40.

13. KONTYNUACJA WALKI

Czas nie mógł uleczyć wszystkich ran Stallmana, ale był jego sprzymierzeńcem.

Nawet po czterech latach krytyka zapisana przez Raymonda w Katedrze i bazarze (The Cathedral and the Bazaar) tkwiła jak zadra w jego duszy. Boczył się także na Linusa Torvaldsa za

zajęcie pozycji najśawniejszego hakera na świecie. Jeszcze dziś wspomina koszulkę typu T-shirt, która zaczął się pojawiać na targach linuksowych w roku 1999. Zaprojektowano ją na wzór plakatu Gwiezdných Wojen - z Torvaldsem wymachującym mieczem świetlnym jak Luke Skywalker i twarzą Stallmana na szczycie korpusu R2D2. Ta koszulka wciąż gra mu na nerwach, nie tylko dlatego, że przedstawia go w roli pomagiera Linusa, lecz z tego powodu, że podnosi Torvaldsa do rangi przywódcy społeczności wolnego oprogramowania i oprogramowania open source, roli, której nawet on sam sobie nie przypisuje. "To śmieszne, że sam Torvalds nie zamierzał dzierżyć tego miecza - mówi z żalem. - Wszyscy uznali go za przywódcę ruchu, a on nie miał ochoty na walkę. Gdzie tu sens?"

Ta niechęć Torvaldsa do "wzięcia w dł on miecza" pozwoliła Stallmanowi zachować pozycję najwyższego arbitra hakerskiej etyki. Pomimo poczucia krzywdy Stallman musi przyznać, że kilka ostatnich lat to niezły czas dla niego samego i jego organizacji. Odsunięty na ubocze przez nieoczekiwany sukces GNU/Linux, zdołał jednak odzyskać inicjatywę. Od stycznia 2000 do grudnia 2001 odwiedził 6 kontynentów, wygłaszając wykłady w krajach, w których sprawy wolnego oprogramowania cieszą się zainteresowaniem, np. Chinach i w Indiach. Nauczył się także budować swoją pozycję poza katedrą wykładowcy, jako osoba współodpowiedzialna za Powszechną Licencję Publiczną (GPL). Latem 2000, gdy powietrze nagle zaczęło uciekać z nadmuchanego w 1999 roku linuksowego balonu giełdowego, Stallman i jego Fundacja Wolnego Oprogramowania odnieśli dwa znaczne sukcesy. W lipcu 2000 Trolltech, norweska firma programistyczna, twórca Qt - wartościowego pakietu narzędzi graficznych dla systemu operacyjnego GNU/Linux - ogłosił a przyjęcie licencji GPL. Kilka tygodni później firma Sun Microsystems, dotychczas próbująca samodzielnie dbać o zachowanie kontroli nad prawami własności swojego oprogramowania open source, w końcu

złagodziła swoje stanowisko i ogłosiła, że jej pakiet aplikacji OpenOffice będzie odtąd chroniony przez podwójną licencję - Lesser GNU Public License (LGPL)² oraz Sun Industry Standards Source License (SISSL)³.

Warto podkreślić, że oba zwycięstwa przysły bez szczególnych starań ze strony Stallmana. W przypadku Trolltech Stallman pełnił jedynie rolę arcykapłana wolnego oprogramowania. W roku 1999 firma wystąpiła z własną licencją, wychodzącą naprzeciw warunkom Fundacji, ale po przeanalizowaniu jej treści Stallman zauważył prawne niezgodności, które mogły uniemożliwić łączenie Qt z programami chronionymi przez GPL. Zmęczony naleganiami Stallmana, zarząd Trolltech'a ostatecznie zdecydował się rozdzielić Qt na dwie wersje chronione licencjami GPL i QPL, co umożliwiło twórcom oprogramowania obejście wytkniętych przez Stallmana problemów niezgodności.

Sun od początku chciał postępować zgodnie z wytycznymi Fundacji Wolnego Oprogramowania. W roku 1999 na O'Reilly Open Source Conference współzałożyciel Sun Microsystems i szef działu badawczego firmy, Bill Joy, obronił "społeczną licencję" (ang. community licence) stanowiącą kompromisowe rozwiązanie o mniejszej mocy. Pozwalała ona użytkownikom na kopiowanie i modyfikowanie oprogramowania będącego własnością Suna, ale nie na pobieranie opłaty za to oprogramowanie, o ile wcześniej nie została zawarta umowa określająca sposób dzielenia zysków. Rok po przemówieniu Joya wiceprezes Sun Microsystems, Marco Boerries, z tej samej trybuny ogłosił kolejny kompromis. Dotyczył on OpenOffice, pakietu aplikacji zaprojektowanych specjalnie do stosowania w systemie operacyjnym GNU/Linux.

"Mogę to wyrazić za pomocą trzech liter - GPL" - powiedział. Jak stwierdził Boerries, decyzja firmy miała niewiele wspólnego ze Stallmanem, a o wiele więcej z rozmachem rozpowszechniania się programów chronionych licencją GPL. "Przede wszystkim różne produkty są atrakcyjne dla różnych

społeczności i licencja chroniąca dany produkt powinna być dostosowana do wymagań danej społeczności - stwierdził Boerries. - A jest oczywiste, że OpenOffice jest produktem dla społeczności GPL-owskiej"4 i 5.

Takie komentarze pokazywał y nierozpoznaną od razu moc GPL, a pośrednio - polityczny geniusz człowieka, który odegrał najważniejszą rolę przy jej tworzeniu. "Nie było na świecie prawnika, który potrafił by naszkicować GPL w tej postaci - twierdzi Eben Moglen, profesor prawa na Columbia University i generalny doradca Fundacji Wolnego Oprogramowania. - A jak widać, to dział a. Dział a dzięki filozofii stworzonej przez Richarda".

Moglen, niegdyś zawodowy programista, śledzi swą współpracę ze Stallmanem pro publico bono sięgającą roku 1990, gdy został poproszony o poradę prawną w sprawie prywatnej. Współpracował wówczas z ekspertem od szyfrowania, Phillipem Zimmermanem, wspierając go w prawnej batalii z National Security Administration. Prośbę Stallmana uznał za wyróżnienie: "Powiedziałem mu, że codziennie używam Emacsa i musiał bym wykonać wielką pracę prawniczą, gdybym chciał spłacić zaciągnięty dług".

Od tego momentu Moglen o wiele dokładniej niż ktokolwiek inny mógł śledzić przeistaczanie się hakerskiej filozofii Stallmana w prawną rzeczywistość. Twierdzi, że Stalman niemal tak samo podchodzi do tekstu prawnego i kodu programu komputerowego. "Jako prawnik muszę powiedzieć, iż pomysł ,aby w dokumencie prawnym wyszukać wszystkie błędy i niedociągnięcia, nie ma wielkiego sensu - mówi Moglen. - W każdym postępowaniu prawnym jest pewna doza niepewności i prawnicy starają się wyciągnąć stąd korzyści dla swego klienta. Cel Richarda był zupełnie inny. On chciał usunąć wszystkie niejasności, co jest z natury rzeczy niemożliwe. Niemożliwe jest także napisanie jednej licencji, która pasowałaby do wszystkich okoliczności we wszystkich systemach prawnych na całym świecie. Ale jeżeli wejdiesz między wrony, musisz krakać tak

jak one; musiał em się dostosować. Zdumiewa elegancja i prostota wyniku. Chyba został o tu osiągnięte wszystko, co był o możliwe do osiągnięcia. Niewiele prac prawniczych może się z tym równać".

Jako osoba odpowiedzialna za realizację zadań zgodnie z planem nakreślonym przez Stallmana, Moglen doskonale rozumie frustrację potencjalnych sojuszników. "Richard jest człowiekiem nieuznającym kompromisów w sprawach, które uważa za fundamentalne - mówi - i z trudem pojmuję pokrętność słów i poszukiwanie dwuznaczności, która okazuje się pożądana przez część społeczeństwa".

Ponieważ Fundacja Wolnego Oprogramowania nie miała a ochoty na rozważanie kwestii leżących poza zagadnienia GNU i GPL, Moglen resztę swej energii poświęcił współpracy z Electronic Frontier Foundation⁶, organizacji dającej wsparcie osobom ostatnio oskarżonym o łamanie praw autorskich, takim jak np. Dymitr Sklarow⁷. W roku 2000 Moglen był także doradcą grupy hakerów zajmujących się programem deCSS deszyfrującym zapis DVD. Moglen nauczył się doceniać upór swojego głównego klienta pomimo jego małomówności. "Zdarzało się, że przychodził em do Richarda, mówiąc 'Musimy to zrobić.

Musimy tamto zrobić. To jest strategicznie ważna sytuacja. Następny ruch powinien być taki. Oto, co musimy zrobić', na co on odpowiadał 'Nie musimy niczego robić. Poczekajmy, aż to, co potrzebne, zostanie wykonane'".

"I wiesz co? - dodaje Moglen. - Zwykle miał rację".

Takim komentarzom przeczy wypowiedź samego Stallmana:

"Nie nadaję się do prowadzenia rozgrywek - odpowiada wielu krytykom uznającym go za przebiegłego egostratega. - Nie potrafię przewidywać działań innych. Zawsze zwracał em uwagę głównie na budowanie podstaw. Mówił em 'Temu, co robimy, dajmy możliwie najmocniejsze podstawy'".

Rosnąca popularność GPL i coraz większa siła przyciągania tej licencji najlepiej świadczyły o jakości fundamentów założonych przez Stallmana i jego kolegów z projektu GNU. Stallman, choć

nie mógł się już uważać za "ostatniego prawdziwego hakera", nadal był uznawany za kogoś, kto sam stworzył etyczne zasady ruchu wolnego oprogramowania. Niezależnie od tego, czy współczesnym hakerom w pełni odpowiadały ramy nakreślone przez Stallmana, dzięki niemu mieli wybór sposobu działania i to stanowiło największą wartość jego spuścizny.

Mówienie o spuściznie Stallmana zdaje się być nieco przedwcześnie. W chwili, gdy piszę te słowa, Stallman ma 48 lat i przed sobą czas, w ciągu którego może powiększyć swe dzieło lub nieco uszczknąć z należnej mu chwały. Nadal samostereująca natura ruchu wolnego oprogramowania skłania do spojrzenia na życie Stallmana ponad codziennymi walkami z przemysłem emprogramistycznym i szukania cech ogólniejszych o znaczeniu historycznym.

Stallman odrzuca wszystkie spekulacje dotyczące jego osoby.

"Nigdy nie potrafił opracować szczegółowego planu, przewidującego, co przyniesie przyszłość" - twierdzi i nieco przedwcześnie proponuje, aby na jego nagrobku wyryć epitafium: "Zawsze był on gotowy do walki, choć nie wiedziałem, dokąd mnie ona zawiedzie".

Jest oczywiste, że, wybierając walkę, odstąpił od siebie wielu ludzi, którzy mogli stać się jego największymi sprzymierzeńcami. Hołdem dla jego prostolinijnej etycznej natury jest fakt, że wielu jego niegdyśszych przeciwników politycznych, gdy ich przyprzeć do muru, potrafi powiedzieć o nim nieco dobrego. Zdziwienie biografów budzi jednakże napięcie między Stallmanem-ideologiem i Stallmanem-genialnym hakerem, jednakże nie ma tu miejsca na rozważanie, jak ludzie będą pamiętali Stallmana, gdy jego już nie będzie".

We wczesnych szkicach tej książki nazwał on ten rozdział "Po 100 latach". Mając nadzieję na obiektywne opisanie Stallmana i jego dzieła, wypytywał on wielu ludzi, osobistości przemysłu programistycznego. Proponowałem im wzniesienie się ponad bieżące wydarzenia i spojrzenie na sprawę z pozycji historyka, który będzie badał ruch wolnego oprogramowania po 100

latach. W chwili obecnej łatwiej było porównania Stallmana do wielu Amerykanów z przeszłości, którzy zajmując marginalną pozycję w życiu współczesnym, z perspektywy historycznej sięgnęli wyżyn uznania. Łatwo przeprowadzić paralelę z Henrym Davidem Thoreau⁸ transcendentalnym filozofem i autorem "Cywilnego nieposłuszeństwa" (ang. On Civil Disobedience) oraz Johnem Muirem⁹, założycielem Sierra Club i propagatorem współczesnego ruchu ochrony środowiska. Łatwo też zauważyć podobieństwo do Williama Jenningsa Bryana¹⁰, znanego także jako "Wielki człowiek z ludu" (ang. Great Commoner), lidera populistycznego ruchu społecznego, wroga monopoli, w swoim czasie człowieka potężnego, którego znaczenie jednakże maleje, gdy spoglądamy nań z perspektywy historii.

Choć nie on pierwszy uznał oprogramowanie za własność powszechną, z powodu GPL Stallman zasłużył sobie na notkę w przyszłych podręcznikach historii. Mając to na względzie, warto przesunąć się w czasie, by spojrzeć na spuściznę Stallmana spoza obecnych ram czasowych. Czy w roku 2102 GPL będzie wciąż coś znaczyć dla programistów, czy pozostanie jedynie wspomnieniem po czymś, co dawno zostało odrzucone na margines historii? A może termin "wolne oprogramowanie" będzie jedynie taką osobliwością jak dziś "wolne srebro" (ang. free silver)¹¹? A może jest inspirującą wytyczną przyszłych politycznych zdarzeń?

Przewidywanie przyszłości jest zajęciem ryzykownym, ale wielu ludzi, proszonych o to, chętnie to ryzyko podejmuje. "Za sto lat Richard i kilku innych będą zasługiwali na o wiele więcej uwagi niż krótki przypisek historyczny - mówi Moglen. - Są na drodze do znalezienia się w centrum uwagi historii".

Wśród tych "kilku innych", których Moglen uznaje za godnych znalezienia się na kartach historii, są John Gilmore, najpierw doradca Stallmana w czasie powstawania DPL, a potem założyciel Electronic Frontier Foundation, a także Theodor Holm Nelson, znany jako Ted Nelson¹², autor wydanej w roku

1982 książki "Literary Machines". Moglen twierdzi, że Stallman, Nelson i Gilmore wnieśli do historii trzy różne i niezależne rozwiązania. Nelson, powszechnie znany jako ten, który ukuł pojęcie "hipertekst", był pierwszym, który przewidział niedogodności wynikające z praw własności informacji w wieku technologii cyfrowych, zaś Gilmore' wi i Stallmanowi zawdzięczamy zidentyfikowanie negatywnych politycznych efektów kontrolowania informacji. Oni też stworzyli organizacje zwalczające te zagrożenia: Gilmore - Fundację Elektronicznego Pogranicza (Electronic Frontier Foundation), zaś Stallman - Fundację Wolnego Oprogramowania (Free Software Foundation). Jednakże, zdaniem Moglena, działania Stallmana mają bardziej osobisty i mniej polityczny charakter.

"Richard jest jedynym, który tak wcześnie zauważył etyczne implikacje ŹniewoleniaŹ oprogramowania - mówi Moglen. - Choć wielu piszących o nim uważa zajęcie się tym problemem za dział anie uboczne lub nawet za jego błąd życiowy, w rzeczywistości wynika to bezpośrednio z jego osobowości". Gilmore, który z mieszanymi uczuciami przyjął umieszczenie go między kapryśnym Nelsonem i drażliwym Stallmanem, jednak przytakuje Moglenowi, pisząc:

Przypuszczam, że teksty Stallmana wytrzymają próbę czasu tak, jak wytrzymał o ją b, co napisał Thomas Jefferson.

Stallman pisze jasno i jasno wyraża swe zasady... To, czy stanie się on równie wpł ywowy jak Jeferson, zależy od tego, czy abstrakcje, które nazywamy "prawami obywatelskimi", staną się za 100 lat ważniejsze od abstrakcji, które nazywamy teraz "oprogramowaniem" i "technicznie narzuconymi ograniczeniami".

Gilmore zwraca uwagę, że dorobkiem Stallmana, którego nie można nie dostrzec, jest wprowadzony w projekcie GNU pionierski model współ tworzenia oprogramowania. Choć czasem w postaci skażonej, zdołał on jednak stać się standardem w przemyśle programistycznym. Gilmore przypuszcza, iż być może wł śnie ten model kooperatywnego

tworzenia programów w przyszłości będzie miał większe znaczenie niż projekt GNU, licencja GPL lub którykolwiek z programów napisanych przez Stallmana:

Przed nastaniem Internetu współpraca na odległość była bardzo utrudniona i ciężko było wspólnie tworzyć programy nawet w zespole, którego członkowie znali się nawzajem i ufali sobie. Richard stworzył model kooperatywnego tworzenia programów także przez grupy złożone z osób nieorganizowanych i rzadko się spotykających. Nie zbudował żadnego z narzędzi służących do takiej współpracy (protokołu TCP, list

e-mailowych, narzędzi diff/patch¹³, plików tar¹⁴, RCS¹⁵ lub CVS lub zdalnego CVS¹⁶), ale zawsze używał wszystkiego, co był odstępne i pomocne w tworzeniu grup programistów, którzy mogli efektywnie współpracować.

Wtórą mu Lawrence Lessig, profesor prawa w Stanford i autor książki roku 2001 *The Future of Ideas* (Przyszłość idei). Jak wielu innych wykładowców akademickich Lessig uważa GPL za główny bastion tak zwanej wspólnoty cyfrowej (ang. digital commons), wielkiej społeczności oprogramowania będącego wspólną własnością oraz standardów sieciowych i komunikacyjnych, które w ciągu ostatnich trzech dekad umożliwiły gwałtowny rozwój Internetu. Raczej nie łączy Stallmana z innymi wielkimi pionierami Internetu jak Vannear Bush¹⁷, Vinton Cerf¹⁸ i J. C. R. Licklider¹⁹, którzy przekonali innych do stosowania technologii komputerowej na szeroką skalę. Uważa, że jego wpływ był bardziej osobisty, introspektywny i niepowtarzalny:

(Stallman) zastąpił debatę o tym, co jest, debatą o tym, co być powinno. Zmusił ludzi do zastanowienia się, jak wysoka jest stawka, i stworzył narzędzia do wprowadzenia swych idei w życie... Nie wiem, na jakim miejscu należy go umieścić obok Cerfa i Licklidera. Wymyślił coś zupełnie innego. Nie jest to jakiś nowy rodzaj kodu lub wyposażenie Internetu w nowe możliwości. To raczej uświadomienie ludziom wartości pewnej

części Internetu. Nie sędę, aby udało nam się znaleźć kogoś tej samej klasy przed nim lub po nim.

Oczywiście, nie wszyscy traktują osiągnięcia Stallmana jako warte upamiętnienia. Eric Raymond, jeden z twórców oprogramowania open source, uważa, że przywódcza rola Stallmana znacznie zmalęła po roku 1996 i oglądając w czarodziejskiej szklanej kuli świat roku 2102, spostrzega zmieszanie sprzecznych ocen:

Myślę, że artefakty Stallmana (GPL, Emacs, GCC) będą uznane za dzieła rewolucyjne, kamienie węgielne świata informacji, ale historia będzie mniej łaskawa dla teorii będących podstawą działania RMS, a całkiem zlekceważy jego pragnienia uczynienia się kultowym przywódcą pewnej grupy.

Sam Stallman również ma mieszane uczucia:

Co za dwadzieścia lat historia powie o projekcie GNU, będzie zależało od tego, kto zwycięży w walce o wolność ludzkiej wiedzy. Jeżeli przegramy, pozostaniemy jedynie przypisem na kartach historii. Jeżeli zwyciężymy, nie jest pewne, czy ludzie będą znali rolę systemu operacyjnego GNU; jeżeli będą nazywać ten system "Linuksem", zbudują fałszywy obraz tego, co się działo i przyczyn, które do tego doprowadziły.

Nawet jeżeli zwyciężymy, to, co będą pisać historycy za sto lat, będzie zależało od tego, kto będzie wówczas dzierżył ster władzy politycznej.

Szukając swego odpowiednika wśród dziewiętnastowiecznych postaci historycznych, Stallman wspomina Johna Browna²⁰, wojowniczego abolicjonistę²¹ uznanego za bohatera po jednej stronie linii Masona-Dixona (Mason Dixon line)²² i za szaleńca po drugiej stronie.

Rewolta, której pragnął John Brown, nigdy nie wybuchła, ale podczas późniejszego procesu udało się mu rozbudzić w społeczeństwie żądanie abolicji. Podczas wojny domowej uznawano go za bohatera, ale w 100 lat później, a nawet przez większość XX wieku, podręczniki historii zwykle przedstawiały go jako szaleńca. W okresie legalnej segregacji, gdy bigoteria nie

przynosiła wstydu, Stany Zjednoczone częściowo zaakceptowały to, co stany południowe miały do powiedzenia na swój temat - wiele nieprawdziwych opinii o wojnie secesyjnej i związanych z nią sprawach przeniknęło wówczas do podręczników historii. Takie porównania świadczą o tym, że sam Stallman postrzega swą obecną pracę jako zjawisko peryferyjne wobec głównego biegu zdarzeń, oraz o podwójnej naturze jego obecnej reputacji. Trudno sobie wyobrazić, by opinia o nim kiedykolwiek sięgnęła takich nizin infamii jak opinia o Johnie Brownie po zakończeniu odbudowy ze zniszczeń wojny secesyjnej, jednakże wojenne analogie w jego wypowiedziach nie stały się zaczynem żadnego buntu i łatwo sobie wyobrazić, że w przyszłości jego idee zostaną zapomniane. Nadając działaniom dotyczącym wolnego oprogramowania postać nie masowego ruchu, lecz raczej ciągu prywatnych potyczek z siłami upowszechniającymi oprogramowanie z prywatnymi prawami własności, Stallman zdaje się stawiać swoich, równie upartych jak on, zwolenników w sytuacji uniemożliwiającej zwycięstwo.

Pomimo wszystko może się to kiedyś okazać jego najtrwalszą spuścizną. Moglen w ciągu ostatniego dziesięciolecia z bliska obserwujący te zdarzenia, ostrzega wszystkich, którzy błędnie chcą uważać osobowość Stallmana za dodatek do jego dokonań. Bez takiej osobowości Stallman nie stworzyłby wielu rzeczy godnych uwagi, twierdzi Moglen, byłby urzędnik Sądu Najwyższego:

Największym człowiekiem, dla którego kiedykolwiek pracowałem, był Thurgood Marshall²³. Wiem, co uczyniło go wielkim. Wiem, dlaczego potrafił zmienić świat zgodnie z własnym wyobrażeniem. Porównywanie ich obu, byłoby nadużyciem, gdyż trudno znaleźć dwie bardziej odmienne postaci. Thurgood Marshall był niezwykłym człowiekiem, reprezentował wyrzuconych poza nawias wobec społeczeństwa, które się przed nimi zamknęło. Jego zdolności i umiejętności były zdolnościami i umiejętnościami społecznymi. Pod wszelkimi innymi względami różnili się, ale pomimo to że

wszystkich, których poznałem, tylko Stallman jest tak samo monolitem wykonanym z jednej bryły, z materii, z której zbudowane są gwiazdy.

Aby sprowadzić ten obraz na ziemię, Moglen wspomina spotkanie ze Stallmanem na wiosnę 2000. W mediach zajmujących się sprawami biznesu odbijało się jeszcze echo sukcesu VA Linux IPO i przy okazji omawiano kilka tematów dotyczących wolnego oprogramowania. W środku tego huraganu tematów i historii wymagających komentarza zasiadł ze Stallmanem do lunchu, czując się jak drobina w oku cyklonu. W ciągu kilku następnych godzin spokojnie omawiali zagadnienia dotyczące jednego tematu: wzmocnienia GPL.

"Siedzieliśmy tam, rozważając, co zrobimy z kilkoma problemami, które pojawiły się we wschodniej Europie, i co pocniemy, gdy problem własności zawartości zacznie zagrażać wolnemu oprogramowaniu - wspomina Moglen. - Podczas rozmowy zastanawiałem się, co mogą o nas myśleć ludzie obok. Widzieli dwóch brodatych anarchistów kreślących plany swych następnych poczynąń. Oczywiście, Richard jak zwykle wyskubywał kosmyki ze swej czupryny, upuszczając je do talerza zupy. Ktokolwiek przysłuchiwał się naszej rozmowie, zapewne doszedł do wniosku, że jesteśmy szaleni, ale wiedziałem, że przy tym stole robimy rewolucję. Mówimy o tym, co się stanie, a ten człowiek sprawi, że to się urzeczywistni." Moglen twierdzi, że ten moment jak nic innego przekonał go o elementarnej prostocie stylu działania Stallmana.

"Zabawne - wspomina Moglen. - Powiedziałem do niego 'Richardzie, wiesz co, ty i ja to tacy dwaj faceci, którzy nie zarobią grosza na tej rewolucji', a potem zapłaciłem za lunch, bo wiedziałem, że on nie ma pieniędzy".

Przypisy

- 1 Oto R2D2 w całej okazałości - przyp. tłum.
- 2 Mniejsza Licencja Publiczna GNU - Powszechnie zostało

przyjęte tłumaczenie General Public Licence GNU (GPL) w postaci: Powszechna Licencja Publiczna GNU, ale nie ma jeszcze ogólnie uznanego tłumaczenia Lesser GNU Public License (LGPL). Na stronach WWW polskiej sekcji Fundacji Wolnego Oprogramowania <http://www.gnu.org/home.pl.html> trwa na ten temat dyskusja, w której proponuje się przetłumaczenie "lesser" jako mniejsza, pomniejsza, poślednia itp. - przyp. tłum.

3 Licencja Standardów Przemysłowych Źródeł Sun - tłumaczenie według strony WWW

<http://www.iidea.pl/prod6.htm> - przyp. tłum

4 Patrz: Marco Boerries, wywiad z autorem (lipiec 2000).

5 Na stronie WWW autor cytuje list, który po ukazaniu się tej książki otrzymał od Frederica Lucifriediego. Autor listu uznał za mylną podaną przez autora informację, że pertraktacje Stallmana z Trolltechem i Sunem przebiegły yglądsko, niemal bez wysiłku. Według Lucifriediego, zwłaszcza spór z Trolltechem był długotrwały i zażarty, co podkreśla szczególnie upór i znaczenie zwycięstwa Stallmana. Uwagi Lucifriediego autor zamierza uwzględnić w kolejnym wydaniu. Cały list Lucifriediego można po angielsku przeczytać na stronie

<http://www.faifzilla.org/errors.html> - przyp. tłum.

6 Na stronach WWW są używane dwie nazwy polskie: Fundacja Elektronicznego Pogranicza i Fundacja Elektronicznej Granicy. Na stronie <http://miastokobiet.art.pl/eff/eff.html> możemy przeczytać podstawowe informacje o tej fundacji, m.in.: "Electronic Frontier Foundation działa od lipca 1990 roku, aby zapewnić przestrzeganie zasad gwarantowanych w Konstytucji i Akcie Praw w nowych technologiach komunikacyjnych. Nasi członkowie dołączają do EFF, aby zapewnić sobie informacje o toczących się debatach, które będą kształtować przyszłość elektronicznej komunikacji." - przyp. tłum.

7 W roku 1998 Kongres uchwalił i prezydent podpisał Digital Millenium Copyright Act (DMCA), akt prawny dostosowujący

zasady prawa autorskiego do wymagań współczesnej technologii. Między innymi za przestępstwo zostało uznane tworzenie i rozpowszechnianie technologii służącej do łamania zabezpieczeń przed kopiowaniem informacji zapisanych w postaci elektronicznej.

Na podstawie tej ustawy w lipcu 2001 roku został aresztowany przez FBI Dymitr Sklarow, pracujący dla rosyjskiej firmy ElcomSoft zdolny programista i kryptolog. Jest on jednym z autorów opublikowanego przez tę firmę programu Advanced eBook Processor, programu potrafiącego czytać dokumenty PDF zabezpieczone w formacie eBook i konwertować je do zwykłego PDF. Program ten Sklarow zaprezentował na konferencji Defcon, po czym w czasie powrotu do domu został na lotnisku zatrzymany przez FBI jako winny naruszenia DMCA.

Więcej informacji na ten temat można znaleźć w artykule dostępnym pod adresem <http://arch.ipsec.pl/snews/346.html>, skąd zaczerpnąłem część podanych w tym przypisie informacji - przyp. tłum.

8 Henry David Thoreau (1817 - 1862), amerykański pisarz i filozof. Główny przedstawiciel romantycznego transcendentalizmu, propagującego idee równości i autonomii moralnej wszystkich ludzi oraz ich niezależności wobec autorytetów i społecznej hierarchii - przyp. tłum.

9 John Muir (1836 - 1914), amerykański podróżnik i przyrodnik. Urodził się w Szkocji, a jego rodzina wyemigrowała do Stanów Zjednoczonych, gdy miał 11 lat. Ukończył studia na Uniwersytecie w Wisconsin, a następnie wyruszył do Meksyku. Był organizatorem wielu wypraw przyrodniczych i jest on w Stanach Zjednoczonych uważany za twórcę ruchu ochrony przyrody. Jego imieniem został nazwany jeden z lodowców w Glacier Bay - przyp. tłum.

10 Bryan William Jennings (1860 - 1925), polityk amerykański, z wykształcenia prawnik. W latach 1896 - 1912 faktyczny przywódca Partii Demokratycznej. W 1901 założył w

Lincoln w Nebrasce wpływowy tygodnik "Commoner" (człowiek z ludu). Przeciwnik udziału Stanów Zjednoczonych w I wojnie światowej. Przeprowadził wiele reform, m.in. przyczynił się do przyznania prawa wyborczego kobietom oraz wprowadzenia podatku dochodowego i prohibicji. Po koniec życia zaangażował się w działalność fundamentalistycznego ruchu religijnego, traktującego dosłownie teksty zawarte w Biblii. W 1925 wystąpił jako oskarżyciel w procesie przeciwko nauczycielowi J.Th. Scopesowi wykładowcaemu teorię ewolucji wbrew prawu stanowemu, zakazującemu rozpowszechniania doktryn sprzecznych z Biblią. Mimo wygranej zmarł kilka dni po procesie, który ujawnił jego całkowitą ignorancję w dziedzinie postępu naukowego. Na podstawie encyklopedii internetowej <http://wiem.onet.pl/>. Więcej informacji na temat Jenningsa i sprawy Scopesa można znaleźć na stronie <http://www.creationism.org.pl/artykuly/scopes1.html> - przyp. tłum.

11 Dotyczy polityki monetarnej USA w drugiej połowie XIX w. i kontrowersji między zwolennikami oparcia narodowej waluty wyłącznie na złocie lub na złocie i srebrze. Więcej informacji można znaleźć w Internecie np. na stronach <http://www.bartleby.com/65/fr/freesilv.html>, <http://projects.vassar.edu/1896/currency.html> oraz po polsku w encyklopedii internetowej <http://wiem.onet.pl/> - przyp. tłum.

12 Theodor Holm Nelson (ur. 1937) - "hipertekstowy populistą", twórca terminu "hipertekst", w 1960 ukończył socjologię na Harvardzie. Próbował wynaleźć system operacji na tekście, pozwalający pisarzom łatwiej poprawiać, porównywać i przerabiać to, co napisali. Próbował tego dokonać na długo przed wynalezieniem procesorów tekstu, trudno zatem się dziwić, że pracy nie ukończył.

Swoje główne koncepcje na temat hipertekstu oraz Xanadu - idealnej przestrzeni literackiej pamięci - zawarł w książkach Literary Machines (Maszyny literackie) i Dream Machines

(Maszyny snów). W Dream Machines opisuje różnorodne odmiany hipertekstu: hipergramy (rozgałęziające się obrazy), hipermapy (złożone z nawarstwionych na siebie, przeźroczystych nakładek) oraz rozgałęziające się filmy - przyp. tłum.

13 Narzędzia do porównywania wersji plików i tworzenia listy różnic ich zawartości - przyp. tłum.

14 Tar (dosłownie TApe Archiver) to uniksowy program do umieszczania grupy plików w jednym pliku, który następnie może zostać skompresowany. Takie skompresowane pliki tar są nazywane tarball (dosłownie kula smoły). Na podstawie <http://pl.wikipedia.org/wiki/Tar> - przyp. tłum.

15 Revision Control System (RCS), czyli system kontroli wersji jest zestawem programów służących do zarządzania rozbudowanymi projektami programistycznymi. Szczególnie takimi, w których uczestniczy wiele osób (niejednocześnie). Działaw środowisku Unix/GNU/Linux. Powstał na wydziale informatyki Uniwersytetu Purdue w Indianie - przyp. tłum

16 Concurrent Versions System (CVS) - system równoległego zarządzania wersjami. Patrz: Gregor N. Purdy CVS. Leksykon kieszonkowy Wydawnictwo Helion - <http://helion.pl/ksiazki/cvslek.htm> - przyp. tłum.

17 Vannevar Bush z wykształcenia inżynier elektryk, w roku 1930 skonstruował pierwszy komputer analogowy. Mając obsesję na temat możliwości nowoczesnej nauki, napisał artykuł, w którym opisał swoje wyobrażenie świata w przyszłości. Opisał tam narzędzie, które nazwał memex, przeznaczone do prywatnej archiwizacji i udostępniania informacji. Uznał, że bez mechanizacji ludzkość nie poradzi sobie z napływającym potokiem informacji. Część jego pomysłu jest uznawana za pierwociny World Wide Web (WWW) - przyp. tłum.

18 Vinton G. Cerf ukończył studia i doktoryzował się (1972 r.) z informatyki w UCLA (University of California, Los Angeles). W roku 1974 wspólnie z Robertem E. Kahnem wymyślił język

pozwalający sieci ARPANET porozumiewać się z innymi eksperymentalnymi sieciami komputerowymi, co później doprowadziło do powstania protokołu komunikacyjnego TCP/IP, umożliwiającego działanie Internetu. Nazwy Internet użył po raz pierwszy w artykule A Protocol for Packet Network Intercommunications opublikowanym w IEEE Transactions on Communication w maju 1974 r. - przyp. tłum.

19 J.C.R. Licklider z Massachusetts Institute for Technology (MIT) w 1962 roku opracował koncepcję opłecenia ziemi globalną siecią telekomunikacyjną - przyp. tłum.

20 Brown John (1800 - 1859), abolicjonista amerykański. W latach 1855 - 1856 organizował zbrojne wypadki w celu uwalniania niewolników w stanie Kansas. W 1859 wraz z grupą zbrojnych zaatakował arsenał w Harpers Ferry w Wirginii w celu wywołania powstania niewolników na Południu. Ujęty, skazany na śmierć i stracony. Jego egzekucja zaoignęła konflikt Północy i Południa na tle niewolnictwa.

Według encyklopedii internetowej <http://wiem.onet.pl/> - przyp. tłum.

21 Ruch mający na celu zniesienie niewolnictwa i handlu niewolnikami, który w XVIII - XIX w. objął Europę i obie Ameryki - przyp. tłum.

22 Linia graniczna między stanami północnymi i południowymi wyznaczona w roku 1767 przez brytyjskich astronomów Charlesa Mason i Jeremiaha Dixona. Stała się później granicą między stanami uznającymi niewolnictwo i stanami dążącymi do jego zniesienia - przyp. tłum.

23 Thurgood Marshall (1908 - 1993) jest jedną z najbardziej znanych postaci amerykańskiej historii walki o prawa obywatelskie czarnej ludności i pierwszym czarnym sędzią Sądu Najwyższego. Funkcję tę pełnił przez 24 lata do 28 czerwca 1991, gdy odszedł z powodu zaawansowanego wieku i pogarszającego się stanu zdrowia - przyp. tłum.

EPILOG MIAŻDŻĄCA SAMOTNOŚĆ

Pisanie biografii osoby żyjącej przypomina wystawianie sztuki w teatrze. Przedstawienie na scenie często błędnie w porównaniu z dramatem za kulisami.

Pisząc "Autobiografię Malcolma X" (The Autobiography of Malcolm X), Alex Haley dał czytelnikom rzadką okazję zajrzenia za kulisy. Wychodząc poza rolę ghostwritera¹, w epilogu przemówił własnym głosem. Wyjaśnił, w jaki sposób niezależny reporter, przegnany przez rzecznika Nation of Islam jako "pachołek" i "szpieg", przeniknął przez polityczne i personalne bariery, aby przelać na papier historię życia Malcolma X.

Waham się, czy mogę porównywać tę książkę z "Autobiografią Malcolma X", ale chciałbym wyrazić głęboką wdzięczność dla Haleya za napisanie tak szczerego i otwartego epilogu. Przez ostatnich 12 miesięcy był on dla mnie podręcznikiem, instrukcją opisywania postaci, która całą karierę zbudowała na sprzeciwie. Dlatego tę biografię postanowiłem zakończyć podobnym epilogiem, który ma być hołdem złożonym Haleyowi i informacją mówiącą czytelnikom, jak ta książka powstawała. Przygoda z pisaniem tej opowieści zaczął się w mieszkaniu w Oakland i prowadził mnie przez miejsca tak różne, jak: Dolina Krzemowa, Maui, Boston i Cambridge. Ostatecznie jest to jednak historia o dwóch miastach: Nowym Jorku, światowej stolicy wydawców książek, i Sebastopolu w Kalifornii, stolicy wydawców Sonoma County.

Wszystko zaczęło się w kwietniu 2000 roku. Pisałem wówczas historię dla złotej sławy witryny WWW BeOpen (<http://www.beopen.com>). Jednym z pierwszych zadań było przeprowadzenie telefonicznego wywiadu z Richardem M. Stallmanem. Wywiad wypadł dobrze, aż tak dobrze, że łącze do niego zostało umieszczone na stronie WWW VA Software, Inc. (niegdyś VA Linux Systems, a poprzednio VA Research). W ciągu kilku godzin serwery BeOpen zagrzały się od licznych kliknięć użytkowników chcących przeczytać ten artykuł.

Wszystkie znaki na niebie i ziemi wskazywały, że na tym się skończy. Jednakże gdy w trzy miesiące później uczestniczyłem w O'Reilly Open Source Conference w Monterey w Kalifornii, otrzymałem e-mail od Tracy Pattison, zarządzającej sprawami obcych praw autorskich w jednym z wielkich wydawnictw nowojorskich:

Do: sam@BeOpen.com

Temat: Wywiad z RMS

Data: 10 lipca 2000, 15:56:37 -0400

Drogi Panie Williams

Z wielkim zainteresowaniem przeczytałem na stronie BeOpen Pański wywiad z Richardem Stallmanem. Od pewnego czasu intryguje mnie RMS i jego działanie i z przyjemnością stwierdzam, że Pański tekst o nim był doskonały. Udało się Panu świetnie przekazać ducha działań Stallmana dotyczących GNU-Linuxa i Fundacji Wolnego Oprogramowania.

Chciałabym jednak dowiedzieć się więcej i sadzę, że nie tylko ja. Czy sądzi Pan, że jest dostępnych więcej źródeł, które pozwoliłyby na rozszerzenie i uaktualnienie Pańskiego wywiadu w celu lepszego nakreślenie sylwetki Stallmana? Dodanie nieco anegdot i informacji zza kulis mogłoby naprawdę zainteresować i poinformować wielu czytelników spoza ścisłego kręgu programistów.

E-mail kończył się prośbą o telefon w celu dokładniejszego omówienia sprawy. Zadzwoniłem. Tracy poinformował mnie, że jej wydawnictwo rozpoczyna wydawanie serii książek elektronicznych i szuka historii, które mogą trafić do czytelników przyzwyczajających się do nowego medium. Format e-booka obejmował 30 tys. słów, czyli około 100 stron. Tracy przedstawił mi szefowi pomysł opisanie głównych postaci hakerskiej społeczności. Szefowi pomysł się spodobał, więc rozpoczęło się poszukiwanie postaci do opisanie. To doprowadziło Tracy do mojego wywiadu ze Stallmanem na stronie BeOpen. Stąd ten e-mail do mnie.

Tracy zapytał, czy zechciałabym rozszerzyć wywiad do pełnego

opisu postaci Stallmana?

Odpowiedziałem natychmiast, że tak. Poprosiła, abym spisał zarys tego, co chcę opisać, aby mogła to przedstawić swoim szefom. Dwa dni później wysłał mi jej żądany opis proponowanego tekstu. Po tygodniu otrzymałem e-mail z odpowiedzią. Jej szefowie "dali zielone światło".

Muszę przyznać, że dopiero wtedy pomyślałem o konieczności pozyskania Stallmana do udziału w tym e-bookowym projekcie. Jako reporter zajmujący się zagadnieniami oprogramowania open source, wiedziałem, że Stallman był pedantem. Do tego czasu dostałem już od niego z pół tuzina e-maili karcących mnie za używanie nazwy "Linux" zamiast "GNU/ Linux". Wiedziałem także, że Stallman szuka sposobów dotarcia ze swoim przesłaniem do szerszego grona odbiorców. Pomyślałem, że jeżeli przedstawię mu projekt od tej strony, być może łatwiej go zaakceptuje. Jeżeli nie, pozostanie mi korzystanie z wielkiej liczby rozsianych w Internecie dokumentów, wywiadów i zapisanych z nim rozmów, a następnie napisanie nieautoryzowanej biografii.

Podczas poszukiwania materiałów trafiłem na esej pod tytułem Freedom-Or Copyright? (Wolność czy prawo autorskie).

Napisany przez Stallmana i opublikowany w czerwcowym wydaniu MIT Technology Review (Przeglądzie technicznym MIT) artykuł nie pozostawiał suchej nitki na e-bookach, wyliczając całą listę ich grzechów. Stallman narzekał, że czytelnicy, aby czytać tekst, muszą korzystać z oprogramowania z zastrzeżonymi prawami własności, a do tego stosuje się bardzo silne zabezpieczenia przed nieautoryzowanym kopiowaniem. Zamiast zwykłego łatwego do przekazywania pliku HTML lub PDF czytelnicy ściągają plik szyfrowany. W rzeczywistości kupowanie e-booka oznacza kupowanie nieprzekazywalnego klucza do rozpakowania zaszyfrowanego pliku. Każda próba odczytania zawartości książki bez autoryzowanego klucza jest uznawana za pogwałcenie Digital Millennium Copyright Act z roku 1998, ustawy stworzonej dla wzmocnienia praw

autorskich w Internecie. Tych, którzy dokonaliby konwersji zawartości książki do pliku z otwartym formatem, czekają kary, nawet gdyby ich jedynym celem było czytanie tekstu na innym domowym komputerze. W przeciwieństwie do zwykłej książki właściciel e-booka nie może go pożyczyć, skopiować lub odsprzedać. Stallman ostrzegał, że może go jedynie czytać na autoryzowanej maszynie:

Używając papierowych książek, zachowujemy wszystkie stare wolności. Sytuacja pogarsza się, gdy książka drukowana zostaje zastąpiona przez elektroniczną. "Elektroniczny atrament" może pisać wciąż na tym samym "papierze", a więc również czasopisma staną się efemerydami. Wyobraźcie sobie: żadnych antykwariatów z używanymi książkami, żadnego pożyczania książek przyjaciołom, żadnego wypożyczania książek z bibliotek publicznych, żadnych "przecieków", pozwalających komuś przeczytać książkę bez płacenia za nią (a uwzględniając dodatki Microsoft Reader, również żadnego anonimowego kupowania książek). Taki świat szykują nam wydawcy².

Nie trzeba chyba wyjaśniać, że takie stawianie sprawy wprowadziło nas w zakłopotanie. Ani Tracy, ani ja nie zastanawialiśmy się, jakiego oprogramowania chce użyć jej firma, i jakie prawa autorskie mają chronić e-bookowe wydanie. Wspomniałem Tracy o artykule w Technology Review i poprosiłem o informacje na temat polityki wydawnictwa dotyczącej e-booków. Obiecała dowiedzieć się.

Paląc się do roboty, postanowiłem tak czy inaczej zatelefonować do Stallmana i wspomnieć mu o e-bookowym pomysle. Gdy o tym usłyszał, natychmiast się zainteresował i zgłosił zastrzeżenia. "Czy czytałś mój artykuł o e-bookach?" - zapytał.

Gdy powiedziałem, że tak i że czekam na informacje z wydawnictwa, postawił dwa warunki: nie będzie wspierał e-bookowego mechanizmu licencjonowania, któremu jest z zasady przeciwny, i nie chce, aby wyglądał o, że ten rodzaj

wydawnictw popiera. "Nie chcę brać udziału w czymkolwiek, co pokazywałoby mnie jako hipokrytę" - stwierdził.

Dla niego samo oprogramowanie było sprawą mniejszej wagi niż sposób jego licencjonowania. Stwierdził, że dopóty jest mu obojętne, jakie oprogramowanie jest używane, dopóki wydawca w licencji praw autorskich umożliwia dosłowne kopiowanie i rozpowszechnianie zawartości e-booka. Wskazał na The Plant Stephena Kinga jako na model możliwy do zastosowania. W czerwcu 2000 King ogłosił na swej oficjalnej stronie WWW, że postanowił samodzielnie wydać The Plant w postaci kolejnych odcinków. Zgodnie z zapowiedzią, książka w całości miała kosztować 13 dolarów podzielonych na ciąg wpłat jednodolarowych. Dopóty, dopóki przynajmniej 75% czytelników będzie płacić za kolejne rozdziały, King będzie udostępniał następne. Do sierpnia wydawało się, że pomysł działa - King opublikował już dwa rozdziały i pracował nad trzecim.

"Można by zaakceptować coś takiego - mówił Stallman, - jeżeli tylko czytelnicy otrzymają prawo sporządzania dokładnych kopii".

Przekazałem tę informację Tracy. Przekonany, że mnie i Tracy uda się dojść do porozumienia, zadzwonił do Stallmana i umówiłem się z nim na pierwszy wywiad związany z tą książką. Zgodził się bez dalszego dopytywania o szczegóły prawne wydania. Wkrótce po pierwszym wywiadzie przeprowadziłem następny (ten w Kihei), zorganizowany przed 14-dniowym urlopem Stallmana na Tahiti.

Podczas wakacji Stallmana z wydawnictwa nadeszły wieści. Dział prawny nie zgadzał się na zmianę noty copyright dla e-booków. Czytelnicy, którzy chcieliby dysponować książką w postaci zdadnej do przekazywania innym, musieliby złamać kod zabezpieczający i dokonać konwersji do otwartego formatu HTML. Oznaczałoby to przestępstwo podlegające karze.

Mając dwa nowe wywiady, resztę książki musiałbym pisać bez dostępu do nowych materiałów. Postanowiłem jak najszybciej

wybrać się do Nowego Jorku na rozmowę z moim agentem i z Tracy, mając nadzieję na znalezienie kompromisowego rozwiązania.

Po przylocie do Nowego Jorku, spotkałem się z moim agentem, Henningiem Guttmanem. Było to nasze pierwsze osobiste spotkanie - Henning stwierdził, że czarno widzi sprawę. Nie wydawało mu się, że uda się przekonać wydawcę do kompromisu. Wielkie domy wydawnicze z podejrzliwością spoglądały na e-booki i nie miały ochoty na żadne ułatwiające unikanie płacności eksperymenty z prawami autorskimi. Jednakże jako agent specjalizujący się w książkach dotyczących technologii, był zaintrygowany naturą moich kłopotów. Powiedział mi o dwóch już odbytych wywiadach i obietnicy niewydawania książki w sposób, który "uczyniłby ze Stallmana hipokrytę". Uznawszy to za zobowiązanie etyczne, postanowił przyjąć to za punkt wyjścia do dalszych negocjacji. Poza tym pozostawało nam zawsze stosowanie metody "kija i marchewki". "Marchewką" byłaby reklama związana z pierwszym w świecie wydaniem e-booka zgodnie z zasadami wewnętrznej etyki hakerskiej. "Kijem" było ryzyko niespełnienia tych zasad. Na dziewięć miesięcy przed słynną w Internecie sprawą Dimitra Sklarowa zdawaliśmy sobie sprawę, że tylko kwestią czasu jest opracowanie przez jakiegoś hakera metody transmisji plików e-bookowych do otwartego formatu. Wiedzieliśmy także, że wydanie chronionego kodem e-booka o Richardzie Stallmanie był równoważne z wydrukowaniem książki z napisem na okładce "Ukradnij mnie". Po rozmowie z Henningiem zatelefonowałem do Stallmana. Chcąc uczynić "marchewkę" bardziej smakowitą, dyskutowałem z nim o dalszych możliwych kompromisach. Co będzie, jeżeli wydawca opublikuje książkę w podwójnej wersji z dwiema różnymi licencjami, tak jak Sun Microsystems postąpił z Open Office, pakietem wolnego oprogramowania biurowego? Wydawca mógłby wydać wersję e-bookową w normalnym formacie, korzystając ze wszystkich standardowych obostrzeń

związanych z oprogramowaniem tego rodzaju, oraz mniej estetyczną, kopiowalną wersję HTML-ową.

Stallman orzekł, że w zasadzie nie sprzeciwia się pomysłowi rozszczepienia licencji, ale nie chce, aby wersja wolna była gorsza od płatnej. Poza tym stwierdził, że pomysł jest dość kłopotliwy. Podwójna licencja jedynie dlatego został a zastosowana do Sunowskiego Open Office, że on nie miał żadnego wpływu na tę decyzję. Jednakże w tym przypadku on, Stallman, może mieć wpływ na wynik, gdyż może odmówić współpracy.

Zasugerowałem kilka innych rozwiązań, ale bez żadnego efektu. Udało mi się jedynie przekonać go, by zawarte w licencji e-booka zezwolenie na udostępnianie ograniczyć jedynie do rozpowszechniania niekomercyjnego.

Na koniec zasugerował, abym zawiadomił wydawcę, iż obiecałem mu, że książka będzie wolno dostępna. Na to nie mogłem przystać, ale stwierdziłem, że praca bez jego współpracy wydaje mi się niemożliwa do skończenia. Wydawał się być usatysfakcjonowany i zakończył rozmowę swym zwyczajowym "Szczęśliwego hakowania".

Ja i Henning spotkaliśmy się z Tracy następnego dnia.

Powiedział a, że firma jest skłonna zgodzić się na wydanie niezakodowanych fragmentów książki, które będzie można swobodnie kopiować, ale ich wielkość ogranicza do 500 słów.

Henning stwierdził, że to nie wystarczy do spełnienia obietnicy złożonej przeze mnie Stallmanowi. Tracy wspomniała o zobowiązaniach jej firmy wobec takich dostawców sieciowych jak Amazon.com. Zgoda na jednorazowe odkrycie zawartości e-booka oznaczałaby ryzyko narażenia się na pretensje partnerów handlowych. Zakładając, że nie zmienią się stanowiska wydawcy i Stallmana, musiałem podjąć decyzję. Mogłem wykorzystać dwa już przeprowadzone wywiady i złamać obietnicę daną Stallmanowi lub postąpić zgodnie z etyką dziennikarską i wycofać się z ustnej obietnicy danej wydawcy, rezygnując z pisania książki.

Po spotkaniu poszedłem z agentem do pubu na Third Avenue. Z jego telefonu komórkowego zadzwonił em do Stallmana, a ponieważ nikt nie odebrał, nagrał em się na automatyczną sekretarkę. Henning zostawił mnie samego, abym miał czas na zebranie myśli. Po chwili wrócił i podając mi telefon komórkowy, powiedział :

"To Stallman"

Od samego początku rozmowa przebiegała marnie.

Przekazał em, co Tracy mówiła o zobowiązaniach zewnętrznych wydawcy.

"Tak - stwierdził Stallman beznamietnie. - A co mnie obchodzą ich przekłete zobowiązania i kontrakty z innymi?"

Stwierdził em, że wymaganie, by jeden z głównych domów wydawniczych ryzykował bitwę prawną z partnerami z powodu zawierającego 30 000 słów e-booka, to zbyt wiele.

"Nie rozumiesz, że właśnie o to chodzi? - powiedział Stallman. - Właśnie po to to robię. Potrzebne jest symboliczne zwycięstwo. Chcę, aby dokonali wyboru między wolnością i biznesem".

Słowa "symboliczne zwycięstwo" odbijały się echem w moje głowie, gdy wzrokiem bezwiednie błądziłem po oknie, przyglądając się pieszym przesuwającym się po chodniku.

Wchodząc do baru, byłem zadowolony, że znajduje się on zaledwie o przecznice od rogu ulicy upamiętnionego w roku 1976 piosenką Ramonesa 53rd and 3rd, którą lubiłem grać, gdy byłem muzykiem. Tak samo jak owa opisana tam "męska prostytutka" czułem, że wszystko, co wcześniej miało sens, właśnie się rozpada. Ironia losu była niemal namacalna. Po tygodniach spokojnego przyglądania się zmaganiom innych sam znalazłem się w sytuacji kogoś usiłującego dokonać rzeczy niemal niemożliwej: namówienia do kompromisu Richarda Stallmana.

Gdy nadal uśmiechając się i chrząkając, starałem się usprawiedliwić stanowisko wydawcy i przekonać Stallmana o mojej sympatii do niego, on, jak drapieżnik wyczuwający krew, zaatakował :

"Ach, to tak? Starasz się mnie przycisnąć do muru? Zamierzasz po prostu przekonać mnie do ich racji?"

Wrócił em do sprawy podwójnej licencji.

"To znaczy licencji" - uciał krótko.

"Tak licencji. Copyright. Prawo autorskie. Cokolwiek" -

powiedział em, czując się jak ranny zwierz zostawiający za sobą krwawy ślad.

"Aha, więc dlaczego po prostu nie robisz tego, co ci mówię" - wrzasnął .

Musiałem do końca wyciągać przeróżne argumenty w obronie stanowiska wydawcy, bo w moich notatkach pozostała wzmianka, że Stallman rzucił na koniec: "A co mi do tego? To, co robia, to diabelstwo, a ja czortów wspierać nie będę.

Żegnam".

Gdy tylko skończył em rozmowę, mój agent wsunął mi w rękę szklankę świeżo nalanego Guinnessa. "Myślę, że to ci się przyda - powiedział ze śmiechem. - Widział em, że pod koniec diabli cię brali".

Rzeczywiście trząsł em się ze złości. I nie mogł em przestać, dopóki nie opróżnił em ponad połowy Guinnessa. Szlag mnie trafił , gdy usłyszał em, że jestem wysłannikiem "diabła". Byłem wściekły tym bardziej, że jeszcze trzy miesiące temu w mieszkaniu w Oakland rozmyślałem, o czym będę teraz pisał . A teraz siedziałem w miejscu, które dotąd znał em jedynie z rockowych piosenek, spotykałem się wydawcami, pił em piwo z agentem, którego do dnia wczorajszego nigdy na oczy nie widział em. Było to tak surrealistyczne jak oglądanie odbicia własnego życia na filmie wyświetlanym wstecz.

Po jakimś czasie mój organizm jednak odreagował i poczucie absurdu zamieniło się w niepohamowany wybuch śmiechu. W oczach mojego agenta zapewne wyglądało to na nerwowe załamanie kolejnego przewrażliwionego autora. A ja po prostu zacząłem doceniać absurdalne piękno mojej sytuacji. Interes wyjdzie lub nie, ale i tak zbierał em materiał na doskonałą historię. Gdy wybuchy mego konwulsyjnego śmiechu

przycichł y, wzniosłem toast.

"Witam na froncie, przyjacielu - powiedziałem, stukając się z agentem kuflami. - "Cieszymy się, miast smucić".

Gdyby to rzeczywiście była sztuka odgrywana na scenie, miałaby swe romantyczne interludia. Wyczerpani napięciem ostatniego spotkania, zostaliśmy zaproszeni przez Tracy na drinka z jej współpracownikami. Opuściliśmy bar na Third Avenue i skierowaliśmy się w stronę East Village, by spotkać się z nimi.

Rozmawiając z Tracy, starannie unikałem tematów biznesowych. Rozmowa była przyjemna, relaksująca. Przed rozstaniem umówiliśmy się na spotkanie następnego wieczoru. I znów wieczór był tak przyjemny, że Stallman i e-book stał się odległym wspomnieniem.

Po powrocie do Oakland dzwoniłem do różnych zaprzyjaźnionych dziennikarzy i znajomych. Opowiadałem o kłopotliwym położeniu, w jakim się znalazłem. Większość wyrzucała mi, że zbyt dużo obiecałem Stallmanowi na wstępnym spotkaniu. Był wykładowcą szkoły dziennikarskiej, radził, bym zignorował komentarz Stallmana na temat "hipokryty" i po prostu napisał, co mam do napisania.

Dziennikarze, którzy znali stallmanowskie podejście do mediów, dawali dowody sympatii, ale jak jeden mąż mówili to samo: sam musisz się z tym uporać.

Postanowiłem odłożyć książkę na drugi plan. Nawet z dwoma dotychczasowymi wywiadami nie posunąłem się zbyt do przodu. Poza tym dawało mi to szansę rozmawiania z Tracy bez wcześniejszego ustalania spraw z Henningiem. Umówiliśmy się na spotkania jeszcze przed Bożym Narodzeniem: raz ona przyleciała na zachodnie wybrzeże, raz ja poleciałem do Nowego Jorku. Przed końcem stycznia spakowałem laptopa, wszystkie notatki i wylądowałem na lotnisku JFK. Pobraliśmy się 11 maja. To sporo jak na jedną nienapisaną książkę.

Latem zacząłem rozmyślać nad notatkami z wywiadów i napisaniem na ich podstawie artykułu. Nie miałem żadnych

wyrzutów sumienia, gdyż ustalając warunki wywiadów, nic nie mówiliśmy o klasycznych mediach drukowanych. Przyznaję, że czułem się swobodniej, gdyż przez kilka miesięcy sprawa w ogóle nie była poruszana. Od czasu naszej rozmowy telefonicznej otrzymałem od Stallmana tylko dwa e-maile. W obu dawał mi burę za użycie terminu "Linux" zamiast "GNU/Linux", co mi się przydarzył ow kilku artykułach pisanych dla magazynu WWW Upside Today. Poza tym miałem spokój. W czerwcu, mniej więcej w tydzień po przemówieniu na Uniwersytecie Nowojorskim, przymierzyłem się do napisania o Stallmanie artykułu uprasowego na około 5000 słów. Tym razem poszła ogłębko. Upływ czasu spowodował, że emocje opadły i pisałem spokojnie.

W lipcu, w rok od otrzymania pierwszego e-maila od Tracy, zadzwonił Henning. Powiedział, że O'Reilly & Associates, wydawnictwo z Sebastopola w Kalifornii, jest zainteresowane biografią Stallmana. Ucieszyło mnie to. Ze wszystkich wydawnictw na świecie O'Reilly, które wydało The Cathedral and the Bazaar Erica Raymonda, zdawało się najlepiej wyczuwać problem, który zniweczył pierwszą próbę napisania książki o Stallmanie. Jako reporter często korzystałem z wydanego przez O'Reilly Open Sources jako źródła informacji historycznych. Wiedziałem, że różne rozdziały tej książki, w tym rozdział napisany przez Stallmana, zostały opublikowane z licencją zezwalającą na redystrybucję. Takie informacje mogły być przydatne, gdyby ponownie doszło do dyskusji na temat wydania elektronicznego.

I rzeczywiście, sprawa wróciła. Od Henninga dowiedziałem się, że wydawnictwo zamierza wydać książkę w dwóch wersjach, klasycznej oraz w nowej serii subskrybowanych książek elektronicznych Safari Tech Books Online. Henning ostrzegł mnie, że licencja użytkownika Safari zawierał pewne ograniczenia³, ale O'Reilly jest skłonne zezwolić użytkownikom na kopiowanie i rozpowszechnianie tekstu niezależnie od rodzaju publikacji. W zasadzie jako autor mogłem wybierać

między Open Publication License i GNU Free Documentation License.

Sprawdził em zawartość, pochodzenie i podteksty obu licencji. Open Publication License (OPL)⁴ daje czytelnikom prawo reprodukcji i rozpowszechniania pracy w całości lub w częściach w dowolnej postaci "fizycznej lub elektronicznej" z warunkiem objęcia kopii tą samą licencją. Pozwala także, pod pewnymi warunkami, na dokonywanie modyfikacji tekstu. Na koniec licencja ta zawiera kilka punktów ograniczeń do wyboru przez autora, który może ograniczyć tworzenie "istotnie zmienionych wersji" i książek pochodnych bez uzyskania jego wcześniejszej zgody.

Natomiast GNU Free Documentation License (GFDL)⁵ pozwala na kopiowanie i rozpowszechnianie dokumentu w dowolnej postaci, jeżeli tylko zachowana zostanie pierwotna licencja. Pozwala także na modyfikowanie dokumentu pod pewnymi warunkami. Jednak w przeciwieństwie do OPL nie zawiera opcji pozwalających autorowi na ograniczanie wprowadzania zmian. Nie zezwala także na odrzucanie przez autora modyfikacji, które mogą prowadzić do powstania produktu konkurencyjnego. Narzuca także określoną postać przedniej i tylnej okładki, jeżeli ktoś inny niż właściciel praw autorskich zamierza opublikować więcej niż 100 egzemplarzy.

Badając licencje, odwiedziłem także witrynę WWW projektu GNU i jej stronę Various Licenses and Comments About Them (Różne licencje i komentarze o nich)⁶ i ⁷. Znalazł em tam krytyczną opinię Stallmana na temat Open Publication License. Dotyczył aona tworzenia wersji zmodyfikowanych i możliwości wybierania przez autora punktów ograniczających wprowadzanie zmian. Według Stallmana, jeżeli autor nie zamierzał wprowadzać żadnych ograniczeń, lepiej było zdecydować się na GFDL, gdyż minimalizował o to ryzyko wynikające z możliwości, jakie dawał y opcje nie wybrane. W obu licencjach zezwolenie na dokonywanie modyfikacji wynikało z ich pochodzenia. W obu przypadkach prawo

modyfikowania tekstu miało dać możliwość poprawiania instrukcji i podręczników i umożliwienie innym korzystania z wprowadzonych poprawek. Ponieważ moja książka nie miała być podręcznikiem, niewiele mnie obchodziły klauzule dotyczące modyfikacji. Zależało o mi jedynie na tym, aby użytkownicy wersji elektronicznej mieli taką samą wolność rozpowszechniania kopii jak ci, którzy kupią książkę drukowaną. Uważając obie licencje za spełniające moje wymagania, podpisałem umowę przysługującą przez wydawnictwo O'Reilly. Jednak nadal intrygował mnie możliwość nieograniczonego modyfikowania tekstu. Podczas moich poprzednich negocjacji z Tracy zastanawiałem się, jakie wpływająca książkę elektroniczną mogą mieć warunki licencji w stylu GPL. Doszedłem do wniosku, że w najgorszym razie taka licencja zapewni niezłą reklamę, zaś w najlepszym - skłoni czytelników do udziału w procesie tworzenia książki. Jako autor miałem ochotę pozwolić ludziom na wnoszenie poprawek, aby tylko moje nazwisko pozostawało na stronie tytułowej. Ciekawiła mnie także możliwość obserwowania ewolucji książki. Wyobrażałem sobie następne wydania jak rodzaj elektronicznej wersji Talmudu, gdzie mój oryginalny tekst, umieszczony w środkowej kolumnie, byłby otoczony objaśniającymi komentarzami umieszczonymi na obu marginesach.

Inspiracją tej wizji był Project Xanadu (<http://www.xanadu.com>), legendarna koncepcja programistyczna Teda Nelsona z roku 1960. Na Open Source Conference, zorganizowanej przez O'Reilly w roku 1999, widziałem pierwszą demonstrację projektu Udanax będącego odrostem Xanadu i to, co widziałem, zachwyciło mnie. Podczas prezentacji dokument pierwotny i pracę pochodną pokazano w dwóch sąsiednich kolumnach zwykłego tekstu. E-bookowa biografia Stallmana nie miała stać się Udanaxem, ale dawała taką możliwość technologiczną, więc nie należało użytkownikom odbierać szansy takiej zabawy⁸. Gdy Laurie Petrycki, moja redaktorka w O'Reilly pozwolił mi na wybieranie między OPL i GFDL, mogłem znów popuścić

wodze fantazji. Do września 2001, gdy podpisałem kontrakt z O'Reilly, stało się jasne, że e-booki się nie przyjęły. Wiele wydawnictw, w tym wydawnictwo Tracy, straciło zainteresowanie ich wydawaniem. Zadziwił o mnie to i zaciekał o. Czy ta forma przetrwałaby, gdyby został użyta nie jako inna postać publikacji lecz jako narzędzie do budowania więzi społecznych?

Po podpisaniu umowy poinformowałem Stallmana, że sprawa książki znów jest aktualna. Wspomniałem, że O'Reilly daje mi możliwość wyboru między Open Publication License i the GNU Free Documentation License. Stwierdziłem, że skłaniam się raczej ku OPL tylko z tego powodu, aby nie dać innym wydawnictwom możliwości wydania tej samej książki w innych okładkach. Stallman na obronę GFDL podał argument, że O'Reilly już kilkakrotnie z tej licencji korzystało.

Zaproponowałem umowę: wybiorę GFDK, jeżeli Stallman zgodzi się na większą liczbę wywiadów i przyczyni się do rozreklamowania książki. Przystał na wywiady, ale zastrzegł, że jego ewentualny udział w reklamowaniu książki będzie zależał od jej treści. Uważając to za uczciwe postawienie sprawy, umówiłem się na wywiad w Cambridge 17 grudnia 2001.

Postarałem się na skoordynowanie terminu wywiadu ze słabnącą podróżą Tracy do Bostonu. Na dwa dni przed podróżą Tracy zaproponowała, aby zaprosić Stallmana na obiad.

"Tak, czy inaczej, to przecież on nas połączył" - stwierdził a.

Wysłałem do niego e-mail z propozycją i natychmiast otrzymałem akceptującą odpowiedź. Po przyjeździe następnego dnia do Bostonu spotkałem się z Tracy w hotelu, gdzie się zatrzymała, i zamówiłem taksówkę do MIT. Gdy znaleźliśmy się na Tech Square, jeszcze przed zapukaniem do drzwi usłyszeliśmy Stallmana zajętego dyskusją.

"Myślę, że nie będziecie mieć nic przeciwko" - powiedział, otwierając drzwi na tyle, abyśmy oboje mogli wejść i zobaczyć osobę, z którą rozmawiał. Była to młoda, około 25-letnia kobieta. Na imię miała Sarah.

"Pozwolił  m sobie zaprosi  kogo  jeszcze do towarzystwa przy obiedzie" - powiedzia  ,zerkaj c na mnie z chytrym u mieszkiem, takim samym, jakim mnie obdarzy  w restauracji w Palo Alto.

Prawd  m wi c, nie by em zbytnio zaskoczony. Ju  kilka tygodni wcze niej przez matk  Stallmana dotar  y do mnie wie ci,  e ma on now  przyjaci lk . "W ostatnim miesi cu razem byli w Japonii, gdzie Richard pojecha  odebra  nagrod  Takeda" - powiedzia  mi pani Lippman⁹.

W drodze do restauracji dowiedzia em si , w jaki spos b Sarah i Richard si  poznali. Ciekawe,  e te okoliczno ci nie by y mi obce. Pracuj c nad powie ci , Sarah us ysza  o Stallmanie i jego interesuj cej osobowo ci. Postanowi  a,  e pos u y jej jako wz r dla jednej z powie ciowych postaci. Chc c dowiedzie  si , jak w rzeczywisto ci wygl da pierwowz r jej bohatera, poprosi  a go o wywiad. Potem wszystko potoczy  si  szybko. Jak powiedzia  a, byli razem od pocz tku roku 2001.

"Szczzerze podziwiam spos b, w jaki Richard zbudowa  ca y ruch polityczny po wie ony zaspokojeniu tego, co przepe nia  ca e jego jestestwo" - powiedzia  a Sarah, wyja niaj c, co j  przyci gn  do Stallmana.

Moja  ona natychmiast spyta  a: "A c   to by o?"

"Mia dz ca samotno  ".

Podczas obiadu panie zaje y si  w  sn  rozmow , ja za  sp dzi em wi kszo  czasu, staraj c si  dociec, czy ostatnie 12 miesi cy nieco zmie czy  Stallmana. Nie spostrzeg em niczego, co mog oby sugerowa  tak  zmian  osobowo ci. Sta  si  mo e nieco bardziej figlarny i jego oczy do   cz sto spoczywa y na piersiach mojej  ony, ale poza tym pozosta  tak samo "k  j cy jak je ". W pewnym momencie moja  ona powiedzia a do   stanowczo "Bro  Bo e" jedynie po to, by us ysze  typowe Stallmanowskie upomnienie:

"Z przykro ci  mus   ci powiedzie ,  e nie ma  adnego Boga". Jednak e gdy po obiedzie Sarah nas opu ci a, obni y  nieco gard . Gdy szli my do pobliskiej ksi garni, przyzna  , e 12

ostatnich miesięcy diametralnie zmienił o jego spojrzenie na życie. "Myślałem, że moim przeznaczeniem jest spędzić życie w samotności - powiedział .- Cieszę się, że się myliłem".

Przed pożegnaniem wręczył mi swoją "rozrywkową" wizytówkę z adresem, telefonem i wyliczeniem ulubionych przyjemności (dobre książki, dobre jedzenie, egzotyczna muzyka i taniec), abym o tym pamiętał ,umawiając się na następny wywiad.

[RysunekE01]

"Rozrywkowa" wizytówka Stallmana wręczona mi podczas wieczornego obiadu

Następnego dnia, po kolejnym posiłku za sumę, która niech pozostanie w niepamięci, zdawał się być w jeszcze bardziej lirycznym nastroju. Wspominając swe debaty z Currier House na temat korzyści i niedogodności, jakie by przyniosła szczepionka dająca nieśmiertelność, wyraził nadzieję, że naukowcy kiedyś znajdą klucz do tej tajemnicy. "Teraz, gdy zażyłem nieco szczęścia, chciałbym mieć go więcej" - powiedział .

Kiedy wspomniałem komentarz Sarah na temat "miażdżącej samotności", stwierdził ,że nie widzi związku między samotnością w sensie fizycznym i duchowym a samotnością hakera. Jednakże, gdy później wróciliśmy do tego tematu, przyznał ,że samotność lub obawa przed wieczną samotnością był a główną siłą apchającą go do działania w pierwszym okresie projektu GNU.

"Moja fascynacja komputerami nie była od czegokolwiek uzależniona - rzekł .- Moje zainteresowanie nie byłoby mniejsze, gdybym był popularny, a kobiety tłoczyłyby się wokół mnie. Jednakże jest prawda, że na mój charakter głęboko wpłynął fakt, że nigdy nie miałem domu, a gdy znalazłem jeden, zaraz go straciłem, gdy znalazłem drugi, rychło legł w gruzach. Tym, który straciłem, była sypialnia w internacie, a tym, który został zburzony, było AI Lab. Poczucie niepewności wynikające z

braku domu i przynależności do jakiejś społeczności był o przemożne. Walczyłem, by odzyskać to, co straciłem."

Po tym wywiadzie bezwiednie wytworzył się we mnie pewne poczucie emocjonalnej symetrii. Słuchając Sarah mówiącej o tym, co ją pociąga w Stallmanie, i słuchając jego opisu uczuć, które pchnęły go do stworzenia ruchu wolnego oprogramowania, rozmyślałem nad tym, co skłoniło mnie do napisania tej książki. Od lipca 2000 nauczyłem się doceniać zarówno uwodzicielską, jak i odpychającą stronę charakteru Stallmana. Jak Eben Moglen przede mną tak i ja uznałem, że błędem jest traktowanie tej osobowości jako istniejącej obok i wyabstrahowanej z ruchu wolnego oprogramowania. Są to rzeczy, zjawiska wzajemnie się kształtujące i często nierozdzielne.

Jestem pewien, że nie każdy czytelnik w tym samym stopniu odczuwa pokrewieństwo duchowe ze Stallmanem, niektórzy po przeczytaniu tej książki zapewne w ogóle się od niego odłączają, ale większość chyba go zaakceptuje. Niewiele jest osób o osobowości tak pełnej humanizmu. Mam szczerą nadzieję, że inni, wspierani licencją GFLD, poczują nieprzepartą ochotę do uzupełnienia tego wstępnego portretu i rzucają nowe światło na osobę Richarda M. Stallmana.

Przypisy

1 Ghostwriter - pisarz widmo - tak w USA nazywają piszących przemówienia dla polityków, "autobiografie" dla bankierów itd. wydawane potem pod nazwiskiem płacących zleceniodawców - przyp. tłum.

2 Patrz: Safari Tech Books Online; Subscriber Agreement: Terms of Service.

<http://safari.oreilly.com/mainhlp.asp?help=service>

3 Patrz: Safari Tech Books Online; Subscriber Agreement: Terms of Service.

<http://safari.oreilly.com/mainhlp.asp?help=service>

- 4 Patrz: The Open Publication License: Draft v1.0 (8 czerwca 1999). <http://opencontent.org/openpub/>
- 5 Patrz: The GNU Free Documentation License: Version 1.1 (marzec 2000), <http://www.gnu.org/copyleft/fdl.html>
- 6 Patrz: <http://www.gnu.org/philosophy/license-list.html>
- 7 Polski odpowiednik pod adresem <http://www.gnu.org/licenses/licenses.pl.html>
- 8 Każdy, kto zechce wprowadzić tę książkę do projektu Udanax, wolnej wersji Xanadu, spotka się z moim entuzjastycznym poparciem. Więcej informacji na temat tej intrygującej technologii można znaleźć na stronie <http://www.udanax.com>.
- 9 Przed podróżą Stallmana do Japonii nie wiedział em, że Fundacja Takeda postanowił anagrodzić Stallmana, Linusa Torvaldsa i Kena Sakamura swoją pierwszą nagrodą za Techno-Entrepreneurial Achievement for Social/Economic Well-Being. Więcej informacji o tej nagrodzie, wynoszącej milion dolarów, można znaleźć na stronie Fundacji Takeda: <http://www.takeda-foundation.jp>.

A TERMINOLOGIA

W większości przypadków terminu GNU/Linux używał em dla oznaczenia wolego systemu operacyjnego, zaś nazwy Linux - dla oznaczenia samego jądra tegoż systemu. Najbardziej zauważalne odstępstwo od tej zasady miało miejsce w rozdziale 9. W jego końcowej części opisywałem pierwszą fazę powstawania Linuksa jako ewolucyjnej odrośli Miniksa. Można bezpiecznie uznać, że w ciągu dwóch pierwszych lat realizacji tego projektu Torvalds i jego koledzy pracowali niezależnie, niewiele zapożyczając ze stworzonego przez Stallmana systemu GNU. Stopniowo zaczęli sięgać po takie zasoby jak GNU C Compiler i GNU Debugger. Ponadto przed rokiem 1993 Stallman również nie naciskał na

informowanie o dokonanych zapożyczeniach.

Niektórzy mogą uznać za arbitralną decyzję o używaniu w książce nazwy GNU/Linux dla późniejszych wersji systemu.

Chcę jednak zaznaczyć, że nie był to wstępny warunek pozyskania współpracy Stallmana przy pracy nad tą książką.

Była to moja samodzielna decyzja wynikająca częściowo z modularnej natury samego systemu operacyjnego i związanej z nim społeczności, a częściowo - z apolitycznej natury nazwy Linux. Używanie nazwy apolitycznej w biografii Richarda Stallmana wydawał o mi się niestosowne.

W ostatniej fazie pracy nad książką, gdy stało się jasne, że wydawcą będzie O'Reilly & Associates, Stallman stwierdził, że używanie nazwy GNU/Linux zamiast Linux jest warunkiem jego pomocy w reklamowaniu książki po jej opublikowaniu. Gdy zostałem o tym poinformowany, przesłałem Stallmanowi do oceny to, co już zostało napisane, nie mając pojęcia, czy uzna, że postawiony warunek został spełniony.

Podobnie rzecz się miała z terminami "free software" (wolne oprogramowanie) i "open source" (oprogramowanie open source). Również w tym przypadku byłem za stosowaniem wymowniejszego politycznie terminu "wolne oprogramowanie", gdy opisywane były programy, które swobodnie można było kopiować i modyfikować. Choć termin "oprogramowanie open source" jest popularniejszy, zdecydowałem się używać go jedynie do opisywania grup i firm będących zwolennikami tej nazwy i tego oprogramowania. Jednakże w pewnych przypadkach obie nazwy są całkowicie równoważne i mogą być stosowane wymiennie - wówczas postępowałem zgodnie z radą Christine Peterson, osoby, której kompetencje w dziedzinie nazw są ogólnie uznane. Peterson pisze: "Termin 'Wolne oprogramowanie' powinien być nadal używany tam, gdzie lepiej odpowiada sytuacji i kontekstowi. 'Open source' przyjęło się, gdyż nowy termin był potrzebny, a nie dlatego że jest idealny".

B HAK, HAKERZY I HAKOWANIE

Aby w pełni zrozumieć słowo "haker", warto prześledzić jego pochodzenie i wieloletnią ewolucję znaczenia.

W New Hacker Dictionary (Nowym słowniku hakerskim), będącym kompendium żargonu programistycznego, znajdujemy dziewięć znaczeń słowa "hak" (hack) i podobną liczbę znaczeń słowa "haker" (hacker). Tamże zamieszczono artykuł, w którym Phil Agre, haker z MIT, ostrzega czytelników, aby nie dali się zwieść pozornej gietkości tego słowa i możliwości stosowania go w wielu sytuacjach. "Słowo 'hak' ma tylko jedno znaczenie - twierdzi Agre - bardzo subtelne i głęboke, niepoddające się dowolnym interpretacjom".

Niezależnie od rozszerzenia czy zawężenia definicji, ślady pochodzenia tego słowa prowadzą do MIT, gdzie stał o się ono popularnym elementem studenckiego żargonu na początku lat pięćdziesiątych. W roku 1990 muzeum MIT zebrało dokumentację prasową dotyczącą hakerstwa. Wynika z niej, że ówczesni studenci używali słowa "hak" w takim znaczeniu jak ich obecni koledzy słowa "goof" (wygłup śmieszny, ale nie złośliwy). Słowem "hak" można było określić na przykład przestawienie komuś auta spod okna, ale nic złośliwego lub powodującego szkodę, jak na przykład uszkodzenie pomnika w kampusie. Słowo "hak" oznaczało działanie nieszkodliwe, dające radość i rozrywkę.

Od rzeczownika "hak" utworzono formę ciągłą "hakowanie" (hacking). W roku 1950 student, który spędził większą część popołudnia na gadaniu przez telefon lub demontowaniu radioodbiornika, mógł swoje działanie określić jako "hakowanie". Teraz powiedziano by, że to "goog", "goofing" lub "goofing off".

W ciągu lat pięćdziesiątych znaczenie słowa "hak" zaczęło się stawać ostrzejsze, bardziej buntownicze. W tym czasie w MIT panowała ostra rywalizacja i słowo "hakowanie" dostosowywało się do tej ogólnej atmosfery. Dotychczas łagodne żarty stawały się sposobem na "wypuszczenie pary", zażranie administracji

na nosie i pozwolenie sobie na kreatywne myślenie wbrew sztywnym regułom studiów. Wielka liczba korytarzy, przejść z instalacjami podziemnymi i innych pomieszczeń opatrzonych napisem "Przejdźcie wzbronione" dawał a studentom olbrzymie możliwości eksploracji. Korzystanie z takich obejść studenci zaczęli nazywać "tunnel hacking" (tunelowe hakowanie lub tunelowe hakerstwo). Na powierzchni podobne możliwości dawał system połączeń telefonicznych. Zainspirowani przypadkowymi zdarzeniami, dzięki uporowi i pilności studenci nauczyli się stosować różne zabawne triki. Wzorując się na klasycznym już terminie "tunnel hacking", nowe działania określili jako "phone hacking" (hakowanie lub hakerstwo telefoniczne).

Połączenie kreatywnej zabawy i pozbawionej ograniczeń eksploracji to podstawa, z której wyrosł y różne mutacje terminu "hakowanie". Pierwsi, którzy w latach sześćdziesiątych w kampusie MIT sami zaczęli się nazywać hakerami komputerowymi, wywodzili się z powstałej w poprzednim dziesięcioleciu grupy studenckiej Tech Model Railroad Club¹. Wewnątrz klubu działał a ściśła elitarna grupa o nazwie Signals and Power (S&P) Committee², zajmująca się systemami obwodów elektrycznych stosowanych w kolejnictwie. System taki był skomplikowanym układem przekaźników i przełączników, przypominającym lokalną sieć telefoniczną kampusu. Był sterowany za pomocą aparatu telefonicznego, na tarczy którego kierujący wykręcał komendy sterujące ruchem pociągów.

Przyszli inżynierowie elektrycy odpowiedzialni za budowanie tego systemu i utrzymanie go w ruchu uznali, że duch ich działania jest zbliżony do hakerstwa telefonicznego (phone hacking). Przejęli termin "hakowanie" i nadali mu większą klarowność. Z punktu widzenia hakerów z S&P użycie jednego mniej przekaźnika do sterowania jakimś odcinkiem oznaczało możliwość użycia go w przyszłości. "Hakowanie" wysubtelniało i awansowało z synonimu bezproduktywnej zabawy do działania

zwiększającego wydajność i efektywność systemu. Wkrótce członkowie komitetu S&P z dumą zaczęli określać jako "hakowanie" wszelkie działania usprawniające system i przekształcające efektywnie obwody. Siebie samych nazywali hakerami.

Skłonność do zajmowania się skomplikowaną elektroniką oraz tradycyjne wśród studentów MIT upodobanie do przedostawania się przez zamknięte drzwi i wszelkie przejścia z napisami "Wstęp wzbroniony" wkrótce doprowadził ypierwszych hakerów do zainteresowania się maszyną, która właśnie pojawił a się w kampusie. Maszyna nazywał a się TX-0 i był a pierwszym dostępnym na rynku komputerem. Przed końcem lat pięćdziesiątym cała klika S&P przewędrował a do pomieszczenia, gdzie umieszczono TX-0, a wraz z nią w tych ścianach zamieszkał duch kreatywności. Nowy świat komputerowego programowania otworzył przed nimi szerokie pole do popisu i stworzył nową mutację słowa "hakowanie". Przestało ono oznaczać zagł ębienie się w niezwykle systemy elektroniczne, a zaczęło funkcjonować jako określenie opisujące stosowanie niestandardowych, dalekich od oficjalności metod programowania. Oznaczało także zwiększanie efektywności i szybkości działania już istniejących programów, co pozwalało na lepsze wykorzystanie zasobów maszyny. Gwoli prawdy, terminem tym określano także pisanie programów służących jedynie rozrywce i zabawie.

Klasycznym przykładem rozszerzania się znaczenia sł owa "hakowanie" była pierwsza gra wideo Gwiezdne wojny.

Stworzona przez hakerów z MIT na początku lat sześćdziesiątych, miał a wszelkie hakerskie atrybuty w tradycyjnym znaczeniu tego słowa. Był a zabawna i działał a losowo, dając coś więcej niż samą rozrywkę kilkunastu pracującym w nocy hakerom, którzy lubili się nią zabawiać. Ponieważ stworzyli ją dla rozrywki, nie widzieli powodów, by zatrzymywać ją wył ącznie dla siebie, i udostępniali ją wszystkim chętnym. Pod koniec lat sześćdziesiątych stanowił a

ona ulubioną zabawę programistów na całym świecie. Komputerowe hakerstwo w końcu lat sześćdziesiątych ożywiało ducha wspólnego odkrywczego działania i poczucia wspólnej własności tworzonych programów, co znacznie odbiegało od atmosfery "hakerstwa tunelowego" i "hakerstwa telefonicznego" z lat pięćdziesiątych. Tamci hakerzy działali pojedynczo lub w małych grupach i w znacznym stopniu opierali działanie na znajomości kampusu, a to ograniczało możliwość szerokiego rozpowszechniania ich osiągnięć. Hakerzy komputerowi przeciwnie - często korzystali z osiągnięć naukowych i działali w dużych grupach, co owocowało wieloma innowacjami. Nie zawsze pozostawali w zgodzie z oficjalnymi naukowcami komputerowymi, ale szybki rozwój ich branży wymuszał współpracę, którą można nazwać symbiozą.

To właśnie legenda doskonałych umiejętności tych pierwszych hakerów sprawiła, że późniejsi programiści, między innymi Richard Stallman, pragnęli stać się godnymi tego dumnego miana. W połowie lat siedemdziesiątych termin "haker" zaczął mieć elitarne konotacje. W ogólnym sensie haker to osoba pisząca programy komputerowe, ale w sensie szczególnym, to ten, kto potrafi je pisać w sposób doskonały. Tak samo słowo "artysta" ma swoje odcienie znaczeniowe. Nazwanie programisty hakerem było oznaką szacunku. Nazywanie samego siebie hakerem świadczyło o wielkiej zarozumiałości. Ta swoboda interpretowania słowa "haker" zaczęła zanikać wraz z upowszechnianiem się komputerów.

Uściśleniu znaczeniowemu terminu "haker" towarzyszyło przybieranie nowych odcieni semantycznych. Aby być hakerem, nie wystarczyło już pisać ciekawe programy, trzeba było robić coś więcej; haker musiał należeć do hakerskiej subkultury i przestrzegać jej zasad honorowych oraz tradycji w taki sam sposób, jak średniowieczni producenci wina byli zobowiązani do postępowania według zasad cechowych. Społeczna struktura hakerska nie była określona tak ściśle jak dawne przepisy cechowe, ale hakerzy z elitarnych instytucji, takich jak MIT,

Stanford i Carnegie Mellon, zaczęli mówić otwarcie o "hakerskiej etyce", niepisanych zasadach kierujących codziennym postępowaniem hakerów. W roku 1984 Steven Levy napisał książkę Hackers (Hakerzy), w której po wielu konsultacjach zapisał w pięciu głównych punktach zasady hakerskiej etyki.

Pod wieloma względami zasady zapisane przez Levy'ego są nadal podstawą hakerskiej kultury. Dziś ten obraz społeczności zorganizowanej na wzór cechowy został zniekształcony przez zalew populistycznych uprzedzeń rozpowszechnianych przez przemysł programistyczny. Na początku lat osiemdziesiątych komputery zaczęły pojawiać się wszędzie, a programiści, którzy niegdyś dostęp do maszyn mogli uzyskać jedynie w instytucjach i firmach najwyższej rangi, teraz dzięki sieci ARPAnet nagle uzyskali możliwość pracy ręką w rękę z grupą najlepszych. Im bardziej współpracowali, tym więcej przesiąkali anarchistyczną filozofią hakerskiej kultury takich ośrodków jak MIT. Po drodze zgubiono, niestety, właściwą MIT zasadę nieszkodzenia, złamano kulturalne tabu. Im młodszy programiści zaczęli doskonalić swe umiejętności, tym więcej tworzono i rozsiewano po świecie wirusów, tym częściej zaczęto włamywać się do systemów komputerów wojskowych, specjalnie wykorzystując takie systemy jak MIT Oz, popularne wejście do ARPAnetu. Z tego powodu słowo "haker" zdewałowowało się, zyskując negatywne punkowe i nihilistyczne znaczenie. Gdy policja i firmy programistyczne zaczęły śledzić i wykrywać przestępstwa komputerowe popełniane przez programistów-renegatów, przyłapani i postawienie w stan oskarżenia bronili się, cytując przydatne dla ich celów fragmenty zasad etyki hakerskiej. Z tego powodu słowo "haker" w prasie i w czasopiśmie zaczęło się pojawiać w wyraźnie negatywnym znaczeniu. Choć takie książki jak Hakerzy uczyniły wiele, by udokumentować oryginalnego ducha eksploracji i innowacji, który był podstawą hakerskiej kultury, dla większości dziennikarzy słowo "haker" zaczęło oznaczać to samo, co

"elektroniczny włamywacz".

Choć hakerzy od niemal 20 lat przeciwstawiają się błędnemu użyciu tego słowa, zadanie to utrudniają wywodzące się z lat pięćdziesiątych skojarzenia z dążeniami rebelianckim i opinii publicznej trudno odróżnić piętnastolatka łamiącego zabezpieczenia zaszyfrowanych współczesnych programów od studenta z lat sześćdziesiątych otwierającego wytrychem zamki i podważającego drzwi, aby uzyskać dostęp do stojącego w biurze komputera. Tak to łamanie prawa przez jednego przyprawia drugiego o ból głowy. Jednakże dawne hakerskie najważniejsze tabu zabraniające złośliwego i szkodliwego działania jest nadal tak silne, że większość hakerów, osoby z premedytacją niszczące systemy zabezpieczeń, kradnące i niszczące dane, określa się mianem krakerów (crackers)³, uznając ich za oddzielną złośliwą hakerską podgrupę.

Nakaz działania niezłośliwego jest głównym ogniwem łączącym hakowanie z początku XXI wieku z hakowaniem z lat pięćdziesiątych XX wieku. Należy zwrócić uwagę, że o ile pojęcie "hakerstwa komputerowego" ewoluowało w czasie ostatnich czterdziestu lat, o tyle pierwotne znaczenie robienia niewinnych żartów i eksploracji podziemnych przejść w MIT pozostało niezmiennie. Pod koniec roku 2000 muzeum MIT, płacąc trybut starej tradycji, zorganizował o wystawę w oddzielnej sali Hall of Hacks. Znalazło się tam wiele fotografii sięgających roku 1920, z których jedna pokazuje atrapę policyjnego radiowozu. W roku 1993 studenci MIT, składając hołd dawnym hakerskim obyczajom, umieścili tę samą atrapę radiowozu z migającymi światłami policyjnymi na głównej kopule Instytutu. W miejscu znaku policyjnego umieszczono napis IHTFP, popularny w MIT akronim mający wiele znaczeń, z których warto przytoczyć jedno odnoszące się do pełnych napięcia lat pięćdziesiątych: "I hate this fucking place." (Nienawidzę tego pieprzonego miejsca). W roku 1990 muzeum użył o tego akronimu jako nazwy kroniki historii hakerstwa. W tej roli akronim oznacza tytuł: The Institute for Hacks Tomfoolery and Pranks⁴. Nowicjusz może

tam znaleźć krótką historię tego, co w MIT łączy się ze słowem "hacking".

Opisując żart z wystawieniem atrapy samochodu policyjnego, reporter "Boston Globe", Randolph Ryan, napisał w roku 1993, że "w kulturze hakerskiej, tak jak w nauce, najwyżej są cenione elegancja i prostota rozwiązania. Hack różnie się od zwykłego szkolnego wygłupu i często wymaga starannego planowania, projektu inżynierskiego i finezji, nie może się też obejść bez ukrytego dowcipu i inwencji. Niepisane prawo głosi, że "hack" powinien być nieszkodliwy i bezpieczny. Dość często hakerzy potem sami usuwają swoje konstrukcje".

Konieczność zamknięcia hakerskiej kultury komputerowej w tych samych etycznych granicach jest zadaniem palącym, lecz niewykonalnym. Choć wiele działań hakerskich aspiruje do tego samego ducha elegancji i prostoty, świat oprogramowania daje mało szans na cofnięcie złej opinii. Łatwo usunąć policyjny samochód umieszczony gdzieś dla żartu, lecz jak usunąć zakorzenione już przekonanie? Stąd upowszechniane ostatnio rozróżnienie między "czarnymi kapeluszami" (ang. black hats) i "białymi kapeluszami" (ang. white hats), czyli hakerami, którzy swe umiejętności wykorzystują do celów destrukcyjnych i złośliwych, i tymi, którzy działają pozytywnie lub przynajmniej informują o swych poczynaniach.

Haker (hacker), dawny wyraz ze studenckiego żargonu o dość niejasnym i niesprecyzowanym znaczeniu, stał się kula bilardową, obiektem zwrotów politycznych i etycznych niuansów. Być może dlatego tak wielu dziennikarzy i programistów lubi go używać. Ale nikt nie potrafi powiedzieć, dokąd ta kula potoczy się dalej.

Przypisy

1 W wolnym przekładzie "Klub modelowania techniki kolejowej" - przyp. tłum.

2 Komitet sygnałów i zasilania - przyp. tłum.

3 Crack między innymi oznacza łamać, rozłupywać - przyp. tłum.

4 Można to przełożyć jako "Instytut kawałów, wyglądów i psikusów", ale z powodu wieloznaczności słów tłumaczeń może być tyle, ile osób tłumaczących - przyp. tłum.

C LICENCJA GNU WOLNEJ DOKUMENTACJI (GNU FREE DOCUMENTATION LICENSE - GFDL)¹

Licencja GNU Wolnej Dokumentacji²

Wersja 1.1, marzec 2000

Copyright (c) 2000 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Zezwala się na kopiowanie i rozpowszechnianie wiernych kopii niniejszego dokumentu licencyjnego, jednak bez prawa wprowadzania zmian.

0. Preambuła

Celem niniejszej licencji jest zagwarantowanie wolnego dostępu do podręcznika, treści książki i wszelkiej dokumentacji w formie pisanej oraz zapewnienie każdemu użytkownikowi swobody kopiowania i rozpowszechniania wyżej wymienionych, z dokonywaniem modyfikacji lub bez, zarówno w celach komercyjnych, jak i niekomercyjnych. Ponadto Licencja ta pozwala przyznać zasługi autorowi i wydawcy przy jednoczesnym zwolnieniu ich z odpowiedzialności za modyfikacje dokonywane przez innych.

Niniejsza Licencja zastrzega też, że wszelkie prace powstałe na podstawie tego dokumentu muszą nosić cechę wolnego dostępu w tym samym sensie co produkt oryginalny. Licencja stanowi uzupełnienie Powszechnej Licencji Publicznej GNU (GNU General Public License), która jest licencją dotyczącą wolnego oprogramowania.

Niniejsza Licencja została opracowana z zamiarem zastosowania jej do podręczników do wolnego oprogramowania, ponieważ wolne oprogramowanie wymaga wolnej dokumentacji:

wolny program powinien być rozpowszechniany z podręcznikami, których dotyczą te same prawa, które wiążą się z oprogramowaniem. Licencja ta nie ogranicza się jednak do podręczników oprogramowania. Można ją stosować do różnych dokumentów tekstowych, bez względu na ich przedmiot oraz niezależnie od tego, czy zostały opublikowane w postaci książki drukowanej. Stosowanie tej Licencji zalecane jest głównie w przypadku prac, których celem jest instruktaż lub pomoc podręczna.

1. Zastosowanie i definicje

Niniejsza Licencja stosuje się do podręczników i innych prac, na których umieszczona jest pochodząca od właściciela praw autorskich informacja, że dana praca może być rozpowszechniana wyłącznie na warunkach niniejszej Licencji. Używane poniżej słowo "Dokument" odnosić się będzie do wszelkich tego typu publikacji. Ich odbiorcy nazywani będą licencjobiorcami.

"Zmodyfikowana wersja" Dokumentu oznacza wszelkie prace zawierające Dokument lub jego część w postaci dosłownej bądź zmodyfikowanej i/lub przełożonej na inny język.

"Sekcją drugorzędą" nazywa się dodatek opatrzony odrębnym tytułem lub sekcję początkową Dokumentu, która dotyczy wyłącznie związku wydawców lub autorów Dokumentu z ogólną tematyką Dokumentu (lub zagadnieniami z nią związanymi) i nie zawiera żadnych treści bezpośrednio związanych z ogólną tematyką (na przykład, jeżeli Dokument stanowi w części podręcznik matematyki, Sekcja drugorzędna nie może wyjaśniać zagadnień matematycznych). Wyżej wyjaśniany związek może się natomiast wyrażać w aspektach historycznym, prawnym, komercyjnym, filozoficznym, etycznym lub politycznym.

"Sekcje niezmiennie" to takie Sekcje drugorzędne, których tytuły są ustalone jako tytuły Sekcji niezmiennych w nacie informującej, że Dokument został opublikowany na warunkach Licencji.

"Treść okładki" to pewne krótkie fragmenty tekstu, które w nocy informującej, że Dokument został opublikowany na warunkach Licencji, są opisywane jako "do umieszczenia na przedniej okładce" lub "do umieszczenia na tylnej okładce". "Jawna" kopia Dokumentu oznacza kopię czytelną dla komputera, zapisaną w formacie, którego specyfikacja jest publicznie dostępna. Zawartość tej kopii może być oglądana i edytowana bezpośrednio za pomocą typowego edytora tekstu lub (w przypadku obrazów złożonych z pikseli) za pomocą typowego programu graficznego lub (w przypadku rysunków) za pomocą ogólnie dostępnego edytora rysunków. Ponadto kopia ta stanowi odpowiednie dane wejściowe dla programów formatujących tekst lub dla programów konwertujących do różnych formatów odpowiednich dla programów formatujących tekst. Kopia spełniająca powyższe warunki, w której jednak

zostały wstawione znaczniki mające na celu utrudnienie dalszych modyfikacji przez czytelników, nie jest Jawna. Kopię, która nie jest "Jawna", nazywa się "Niejawną".

Przykładowe formaty kopii Jawnych to: czysty tekst ASCII bez znaczników, format wejściowy Texinfo, format wejściowy LaTeX, SGML lub XML wykorzystujące publicznie dostępne DTD, standardowy prosty HTML przeznaczony do ręcznej modyfikacji. Formaty niejawne to na przykład PostScript, PDF, formaty własne, które mogą być odczytywane i edytowane jedynie przez własne edytory tekstu, SGML lub XML, dla których DTD i/lub narzędzia przetwarzające nie są ogólnie dostępne, oraz HTML wygenerowany maszynowo przez niektóre procesory tekstu jedynie w celu uzyskania danych wynikowych.

"Strona tytułowa" oznacza, w przypadku książki drukowanej, samą stronę tytułową oraz kolejne strony zawierające informacje, które zgodnie z tą Licencją muszą pojawić się na stronie tytułowej. W przypadku prac w formatach nieposiadających strony tytułowej "Strona tytułowa" oznacza

tekst pojawiający się najbliżej tytułu pracy, poprzedzający początek tekstu głównego.

2. Kopiowanie dosłowne

Licencjobiorca może kopiować i rozprowadzać Dokument komercyjnie lub niekomercyjnie, w dowolnej postaci, pod warunkiem zamieszczenia na każdej kopii Dokumentu treści Licencji informacji o prawie autorskim oraz noty mówiącej, że do Dokumentu ma zastosowanie niniejsza Licencja, a także pod warunkiem nieumieszczania żadnych dodatkowych ograniczeń, które nie wynikają z Licencji. Licencjobiorca nie ma prawa używać żadnych technicznych metod pomiarowych utrudniających lub kontrolujących czytanie lub dalsze kopiowanie utworzonych i rozpowszechnianych przez siebie kopii. Może jednak pobierać opłaty za udostępnianie kopii. W przypadku dystrybucji dużej liczby kopii Licencjobiorca jest zobowiązany przestrzegać warunków wymienionych w punkcie 3.

Licencjobiorca może także wypożyczać kopie na warunkach opisanych powyżej, a także wystawiać je publicznie.

3. Kopiowanie ilościowe

Jeżeli Licencjobiorca publikuje drukowane kopie Dokumentu w liczbie większej niż 100, a licencja Dokumentu wymaga umieszczenia Treści okładki, należy dołączyć kopie okładek, które zawierają całą wyraźną i czytelną Treść okładki: treść przedniej okładki, na przedniej okładce, a treść tylnej okładki, na tylnej okładce. Obie okładki muszą też jasno i czytelnie informować o Licencjobiorcy jako wydawcy tych kopii. Okładka przednia musi przedstawiać pełny tytuł; wszystkie słowa muszą być równie dobrze widoczne i czytelne. Licencjobiorca może na okładkach umieszczać także inne informacje dodatkowe.

Kopiowanie ze zmianami ograniczonymi do okładek, dopóki nie narusza tytułu Dokumentu i spełnia opisane warunki, może być traktowane pod innymi względami jako kopiowanie dosłowne.

Jeżeli napisy wymagane na którejś z okładek są zbyt obszerne,

by mógł ypozostać czytelne po ich umieszczeniu, Licencjobiorca powinien umieścić ich początek (taką ilość, jaka wydaje się rozsądna) na rzeczywistej okładce, a pozostałą część na sąsiednich stronach.

W przypadku publikowania lub rozpowszechniania Niejawnych kopii Dokumentu w liczbie większej niż 100, Licencjobiorca zobowiązany jest albo dołączyć do każdej z nich Jawną kopię czytelną dla komputera, albo wymienić w lub przy każdej kopii Niejawnej publicznie dostępną w sieci komputerowej lokalizację pełnej kopii Jawnej Dokumentu, bez żadnych informacji dodanych - lokalizację, do której każdy użytkownik sieci miałby bezpłatny anonimowy dostęp za pomocą standardowych publicznych protokołów sieciowych. W przypadku drugim Licencjobiorca musi podjąć odpowiednie środki ostrożności, by wymieniona kopia Jawna pozostała dostępna we wskazanej lokalizacji przynajmniej przez rok od momentu rozpowszechnienia ostatniej kopii Niejawnej (bezpośredniego lub przez agentów albo sprzedawców) danego wydania. Zaleca się, choć nie wymaga, aby przed rozpoczęciem rozpowszechniania dużej liczby kopii Dokumentu Licencjobiorca skontaktował się z jego autorami celem uzyskania uaktualnionej wersji Dokumentu.

4. Modyfikacje

Licencjobiorca może kopiować i rozpowszechniać Zmodyfikowaną wersję Dokumentu na zasadach wymienionych powyżej w punkcie 2. i 3. pod warunkiem ścisłego przestrzegania niniejszej Licencji. Zmodyfikowana wersja pełni wtedy rolę Dokumentu, a więc Licencja dotycząca modyfikacji i rozpowszechniania Zmodyfikowanej wersji przenoszona jest na każdego, kto posiada jej kopię. Ponadto Licencjobiorca musi w stosunku do Zmodyfikowanej wersji spełnić następujące wymogi:

A.

Użyć na Stronie tytułowej (i na okładkach, o ile istnieją) tytułu

innego niż tytuł Dokumentu i innego niż tytuł poprzednich wersji (które, o ile istniały, powinny zostać wymienione w Dokumentcie, w sekcji Historia). Tytuł u jednej z ostatnich wersji Licencjobiorca może użyć, jeżeli jej wydawca wyrazi na to zgodę.

B.

Wymienić na Stronie tytułowej, jako autorów, jedną lub kilka osób albo jednostek odpowiedzialnych za autorstwo modyfikacji Zmodyfikowanej wersji, a także przynajmniej pięciu spośród pierwotnych autorów Dokumentu (wszystkich, jeśli było ich mniej niż pięciu).

C.

Umieścić na Stronie tytułowej nazwę wydawcy Zmodyfikowanej wersji.

D.

Zachować wszelkie noty o prawach autorskich zawarte w Dokumentcie.

E.

Dodać odpowiednią notę o prawach autorskich dotyczących modyfikacji obok innych not o prawach autorskich.

F.

Bezpośrednio po notach o prawach autorskich, zamieścić notę licencyjną zezwalającą na publiczne użytkowanie Zmodyfikowanej wersji na zasadach niniejszej Licencji w postaci podanej w Załączniku poniżej.

G.

Zachować w notce licencyjnej pełną listę Sekcji niezmiennych i wymaganych Treści okładki podanych w notce licencyjnej Dokumentu.

H.

Dołączyć niezmienną kopię niniejszej Licencji.

I.

Zachować sekcję zatytułowaną "Historia" oraz jej tytuł i dodać do niej informację dotyczącą przynajmniej tytułu, roku

publikacji, nowych autorów i wydawcy Zmodyfikowanej wersji zgodnie z danymi zamieszczonymi na Stronie tytułowej. Jeżeli w Dokumencie nie istnieje sekcja pod tytułem "Historia", należy ją utworzyć, podając tytuł, rok, autorów i wydawcę Dokumentu zgodnie z danymi zamieszczonymi na stronie tytułowej, a następnie dodając informację dotyczącą Zmodyfikowanej wersji, jak opisano w poprzednim zdaniu.

J.

Zachować wymienioną w Dokumencie (jeśli taka istniała) informację o lokalizacji sieciowej, publicznie dostępnej Jawnej kopii Dokumentu, a także o podanych w Dokumencie lokalizacjach sieciowych poprzednich wersji, na których został on oparty. Informacje te mogą się znajdować w sekcji "Historia". Zezwala się na pominięcie lokalizacji sieciowej prac, które zostały wydane przynajmniej cztery lata przed samym Dokumentem, a także tych, których pierwotny wydawca wyraża na to zgodę.

K.

W każdej sekcji zatytułowanej "Podziękowania" lub "Dedykacje" zachować tytuł i treść, oddając również ton każdego z podziękowań i dedykacji.

L.

Zachować wszelkie Sekcje niezmiennie Dokumentu w niezminionej postaci (dotyczy zarówno treści, jak i tytułu). Numery sekcji i równoważne im oznaczenia nie są traktowane jako należące do tytułów sekcji.

M.

Usunąć wszelkie sekcje zatytułowane "Adnotacje". Nie muszą one być załączane w Zmodyfikowanej wersji.

N.

Nie nadawać żadnej z istniejących sekcji tytułu "Adnotacje" ani tytułu pokrywającego się z jakkolwiek Sekcją niezmienną. Jeżeli Zmodyfikowana wersja zawiera nowe sekcje początkowe lub dodatki stanowiące Sekcje drugorzędne i niezawierające materiału skopiowanego z Dokumentu, Licencjobiorca może je

lub ich część oznaczyć jako sekcje niezmiennie. W tym celu musi on dodać ich tytuł y do listy Sekcji niezmiennych zawartej w nocie licencyjnej Zmodyfikowanej wersji. Tytuł y te muszą być różne od tytułów pozostał ych sekcji.

Licencjobiorca może dodać sekcję "Adnotacje", pod warunkiem, że nie zawiera ona żadnych treści innych niż adnotacje dotyczące Zmodyfikowanej wersji - mogą to być na przykład stwierdzenia o recenzji koleżeńskie albo o akceptacji tekstu przez organizację jako autorytatywnej definicji standardu.

Na końcu listy Treści okładki w Zmodyfikowanej wersji Licencjobiorca może dodać fragment "do umieszczenia na przedniej okładce" o długości nieprzekraczającej pięciu sł ów, a także fragment o długości do 25 sł ów "do umieszczenia na tylnej okładce". Przez każdą jednostkę (lub na mocy ustaleń przez nią poczynionych) może zostać dodany tylko jeden fragment z przeznaczeniem na przednią okładkę i jeden z przeznaczeniem na tylną. Jeżeli Dokument zawiera już treść okładki dla danej okładki, dodaną uprzednio przez Licencjobiorcę lub w ramach ustaleń z jednostką, w imieniu której działa Licencjobiorca, nowa treść okładki nie może zostać dodana. Dopuszcza się jednak zastąpienie poprzedniej treści okładki nową pod warunkiem wyraźnej zgody poprzedniego wydawcy, od którego stara treść pochodzi.

Niniejsza Licencja nie oznacza, iż autor (autorzy) i wydawca (wydawcy) wyrażają zgodę na publiczne używanie ich nazwisk w celu zapewnienia autorytetu jakiejkolwiek Zmodyfikowanej wersji.

5. Łączenie dokumentów

Licencjobiorca może ł ączyć Dokument z innymi dokumentami wydanymi na warunkach niniejszej Licencji, na warunkach podanych dla wersji zmodyfikowanych w części 4. powyżej, jednak tylko wtedy, gdy w połączeniu zostaną zawarte wszystkie Sekcje niezmiennie wszystkich oryginalnych dokumentów w postaci niezmodyfikowanej i gdy będą one wymienione jako Sekcje niezmiennie połączenia w jego nocie

licencyjnej.

Połączenie wymaga tylko jednej kopii niniejszej Licencji, a kilka identycznych Sekcji niezmiennych może zostać zastąpionych jedną. Jeżeli istnieje kilka Sekcji niezmiennych o tym samym tytule, ale różnej zawartości, Licencjobiorca jest zobowiązany uczynić tytuł każdej z nich unikalnym poprzez dodanie na jego końcu, w nawiasach, nazwy oryginalnego autora lub wydawcy danej sekcji, o ile jest znany, lub unikalnego numeru. Podobne poprawki wymagane są w tytułach sekcji na liście Sekcji niezmiennych w nocie licencyjnej połączenia.

W połączeniu Licencjobiorca musi zawrzeć wszystkie sekcje zatytułowane "Historia" z dokumentów oryginalnych, tworząc jedną sekcję "Historia". Podobnie ma postąpić z sekcjami "Podziękowania" i "Dedykacje". Wszystkie sekcje zatytułowane "Adnotacje" należy usunąć.

6. Zbiory dokumentów

Licencjobiorca może utworzyć zbiór składający się z Dokumentu i innych dokumentów wydanych zgodnie z niniejszą Licencją i zastąpić poszczególne kopie Licencji pochodzące z tych dokumentów jedną kopią dołączoną do zbioru, pod warunkiem zachowania zasad Licencji dotyczących kopii dosłownych we wszelkich innych aspektach każdego z dokumentów.

Z takiego zbioru Licencjobiorca może wyodrębnić pojedynczy dokument i rozpowszechniać go niezależnie na zasadach niniejszej Licencji, pod warunkiem zamieszczenia w wyodrębnionym dokumencie kopii niniejszej Licencji oraz zachowania zasad Licencji we wszystkich aspektach dotyczących dosłownej kopii tego dokumentu.

7. Zestawienia z pracami niezależnymi

Kompilacja Dokumentu lub jego pochodnych z innymi oddzielnymi i niezależnymi dokumentami lub pracami nie jest uznawana za Zmodyfikowaną wersję Dokumentu, chyba że odnoszą się do niej jako do całości prawa autorskie. Taka kompilacja jest nazywana zestawieniem, a niniejsza Licencja

nie dotyczy samodzielnych prac skompilowanych z Dokumentem, jeśli nie są to pochodne Dokumentu. Jeżeli do kopii Dokumentu odnoszą się wymagania dotyczące Treści okładki wymienione w części 3. i jeżeli Dokument stanowi mniej niż jedną czwartą całości zestawienia, Treść okładki Dokumentu może być umieszczona na okładkach zamykających Dokument w obrębie zestawienia. W przeciwnym razie Treść okładki musi się pojawić na okładkach całego zestawienia.

8. Tłumaczenie

Tłumaczenie jest uznawane za rodzaj modyfikacji, a więc Licencjobiorca może rozpowszechniać tłumaczenia Dokumentu na zasadach wymienionych w punkcie 4. Zastąpienie Sekcji niezmiennych ich tłumaczeniem wymaga specjalnej zgody właścicieli prawa autorskiego. Dopuszcza się jednak zamieszczanie tłumaczeń wybranych lub wszystkich Sekcji niezmiennych obok ich wersji oryginalnych. Podanie tłumaczenia niniejszej Licencji możliwe jest pod warunkiem zamieszczenia także jej oryginalnej wersji angielskiej. W przypadku niezgodności pomiędzy zamieszczonym tłumaczeniem a oryginalną wersją angielską niniejszej Licencji moc prawną ma oryginalna wersja angielska.

9. Wygaśnięcie

Poza przypadkami jednoznacznie dopuszczonymi na warunkach niniejszej Licencji nie zezwala się Licencjobiorcy na kopiowanie, modyfikowanie, czy rozpowszechnianie Dokumentu ani też na cedowanie praw licencyjnych. We wszystkich pozostałych wypadkach każda próba kopiowania, modyfikowania lub rozpowszechniania Dokumentu albo cedowania praw licencyjnych jest nieważna i powoduje automatyczne wygaśnięcie praw, które licencjobiorca nabył z tytułu Licencji. Niemniej jednak w odniesieniu do stron, które już otrzymały od Licencjobiorcy kopie albo prawa w ramach niniejszej Licencji, licencje nie zostaną anulowane, dopóki strony te w pełni się do nich stosują.

10. Przyszłe wersje Licencji

W miarę potrzeby Free Software Foundation może publikować nowe poprawione wersje GNU Free Documentation License. Wersje te muszą pozostawać w duchu podobnym do wersji obecnej, choć mogą się różnić w szczegółach dotyczących nowych problemów czy zagadnień. Patrz <http://www.gnu.org/copyleft/>.

Każdej wersji niniejszej Licencji nadaje się wyróżniający ją numer. Jeżeli w Dokumencie podaje się numer wersji Licencji, oznaczający, iż odnosi się do niego podana "lub jakakolwiek późniejsza" wersja licencji, Licencjobiorca ma do wyboru stosować się do postanowień i warunków albo tej wersji, albo którejkolwiek wersji późniejszej opublikowanej oficjalnie (nie jako propozycja) przez Free Software Foundation. Jeśli Dokument nie podaje numeru wersji niniejszej Licencji, Licencjobiorca może wybrać dowolną wersję kiedykolwiek opublikowaną (nie jako propozycja) przez Free Software Foundation.

Załącznik: Jak zastosować tę Licencję dla swojego dokumentu
Aby zastosować tę Licencję w stosunku do dokumentu swojego autorstwa, dołącz kopię Licencji do dokumentu i zamieść następującą informację o prawach autorskich i uwagi o licencji bezpośrednio po stronie tytułowej.

Copyright (c) ROK TWOJE IMIE I NAZWISKO

Udziela się zezwolenia do kopiowania rozpowszechniania i/lub modyfikację tego dokumentu zgodnie z zasadami Licencji GNU Wolnej Dokumentacji w wersji 1.1 lub dowolnej późniejszej opublikowanej przez Free Software Foundation; wraz z zawartymi Sekcjami Niezmiennymi LISTA TYTUŁÓW SEKCJI, wraz z Tekstem na Przedniej Okładce LISTA i z Tekstem na Tylnej Okładce LISTA. Kopia licencji załączona jest w sekcji zatytułowanej "GNU Free Documentation License"

Jeśli nie zamieszczasz Sekcji Niezmiennych, napisz "nie zawiera Sekcji Niezmiennych" zamiast spisu sekcji niezmiennych. Jeśli nie umieszczasz Tekstu na Przedniej Okładce, wpisz "bez Tekstu

na Okładce" w miejsce "wraz z Tekstem na Przedniej Okładce LISTA", analogicznie postap z "Tekstem na Tylnej Okładce" Jeśli w twoim dokumencie zawarte są nieszablonowe przykłady kodu programu, zalecamy, abyś także uwolnił te przykłady, wybierając licencję wolnego oprogramowania, taką jak Powszechna Licencja Publiczna GNU, w celu zapewnienia możliwości ich użycia w wolnym oprogramowaniu.

GNU Free Documentation License (GFDL)

GNU Free Documentation License Version 1.1, March 2000

Copyright (C) 2000 Free

Software Foundation, Inc. 59 Temple Place, Suite 330,
Boston, MA 02111-1307

USA Everyone is permitted to copy and distribute verbatim
copies of this

license document, but changing it is not allowed.

PREAMBLE

The purpose of this License is to make a manual, textbook, or
other written

document "free" in the sense of freedom: to assure everyone
the effective

freedom to copy and redistribute it, with or without modifying
it, either

commercially or noncommercially. Secondly, this License
preserves for the

author and publisher a way to get credit for their work, while
not being

considered responsible for modifications made by others.

This License is a kind of "copyleft," which means that
derivative works of the

document must themselves be free in the same sense. It
complements the GNU

General Public License, which is a copyleft license designed
for free

software.

We have designed this License in order to use it for manuals for free

software, because free software needs free documentation: a free program

should come with manuals providing the same freedoms that the software does.

But this License is not limited to software manuals; it can be used for any

textual work, regardless of subject matter or whether it is published as a

printed book. We recommend this License principally for works whose purpose is

instruction or reference.

APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed

by the copyright holder saying it can be distributed under the terms of this

License. The "Document", below, refers to any such manual or work. Any member

of the public is a licensee, and is addressed as "you."

A "Modified Version" of the Document means any work containing the Document or

a portion of it, either copied verbatim, or with modifications and/or

translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the

Document that deals exclusively with the relationship of the publishers or

authors of the Document to the Document's overall subject (or to related

matters) and contains nothing that could fall directly within that overall

subject. (For example, if the Document is in part a textbook of

mathematics, a

Secondary Section may not explain any mathematics.) The relationship could be

a matter of historical connection with the subject or with related matters, or

of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are

designated, as being those of Invariant Sections, in the notice that says that

the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as

Front-Cover Texts or Back-Cover Texts, in the notice that says that the

Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy,

represented in a format whose specification is available to the general

public, whose contents can be viewed and edited directly and straightforwardly

with generic text editors or (for images composed of pixels) generic paint

programs or (for drawings) some widely available drawing editor, and that is

suitable for input to text formatters or for automatic translation to a

variety of formats suitable for input to text formatters. A copy made in an

otherwise Transparent file format whose markup has been designed to thwart or

discourage subsequent modification by readers is not

Transparent. A copy that

is not "Transparent" is called "Opaque."

Examples of suitable formats for Transparent copies include plain ASCII

without markup, Texinfo input format, LaTeX input format, SGML or XML using a

publicly available DTD, and standard-conforming simple HTML designed for human

modification. Opaque formats include PostScript, PDF, proprietary formats that

can be read and edited only by proprietary word processors, SGML or XML for

which the DTD and/or processing tools are not generally available, and the

machine-generated HTML produced by some word processors for output purposes

only.

The "Title Page" means, for a printed book, the title page itself, plus such

following pages as are needed to hold, legibly, the material this License

requires to appear in the title page. For works in formats which do not have

any title page as such, "Title Page" means the text near the most prominent

appearance of the work's title, preceding the beginning of the body of the

text.

VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or

noncommercially, provided that this License, the copyright notices, and the

license notice saying this License applies to the Document are reproduced in

all copies, and that you add no other conditions whatsoever to those of this

License. You may not use technical measures to obstruct or control the reading

or further copying of the copies you make or distribute.

However, you may

accept compensation in exchange for copies. If you distribute a large enough

number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may

publicly display copies.

COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the

Document's license notice requires Cover Texts, you must enclose the copies in

covers that carry, clearly and legibly, all these Cover Texts:
Front-Cover

Texts on the front cover, and Back-Cover Texts on the back cover. Both covers

must also clearly and legibly identify you as the publisher of these copies.

The front cover must present the full title with all words of the title

equally prominent and visible. You may add other material on the covers in

addition. Copying with changes limited to the covers, as long as they preserve

the title of the Document and satisfy these conditions, can be treated as

verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you

should put the first ones listed (as many as fit reasonably) on the actual

cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than

100, you must either include a machine-readable Transparent copy along with

each Opaque copy, or state in or with each Opaque copy a publicly-accessible

computer-network location containing a complete Transparent copy of the

Document, free of added material, which the general network-using public has

access to download anonymously at no charge using public-standard network

protocols. If you use the latter option, you must take reasonably prudent

steps, when you begin distribution of Opaque copies in quantity, to ensure

that this Transparent copy will remain thus accessible at the stated location

until at least one year after the last time you distribute an Opaque copy

(directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the

Document well before redistributing any large number of copies, to give them a

chance to provide you with an updated version of the Document.

MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the

conditions of sections 2 and 3 above, provided that you

release the Modified

Version under precisely this License, with the Modified Version filling the

role of the Document, thus licensing distribution and modification of the

Modified Version to whoever possesses a copy of it. In addition, you must do

these things in the Modified Version:

Use in the Title Page (and on the covers, if any) a title distinct from that

of the Document, and from those of previous versions (which should, if there

were any, be listed in the History section of the Document).

You may use the

same title as a previous version if the original publisher of that version

gives permission.

List on the Title Page, as authors, one or more persons or entities

responsible for authorship of the modifications in the Modified Version,

together with at least five of the principal authors of the Document (all of

its principal authors, if it has less than five).

State on the Title page the name of the publisher of the Modified Version,

as the publisher.

Preserve all the copyright notices of the Document.

Add an appropriate copyright notice for your modifications adjacent to the

other copyright notices.

Include, immediately after the copyright notices, a license notice giving

the public permission to use the Modified Version under the terms of this

License, in the form shown in the Addendum below.

Preserve in that license notice the full lists of Invariant Sections and

required Cover Texts given in the Document's license notice. Include an unaltered copy of this License.

Preserve the section entitled "History," and its title, and add to it an

item stating at least the title, year, new authors, and publisher of the

Modified Version as given on the Title Page. If there is no section entitled

"History" in the Document, create one stating the title, year, authors, and

publisher of the Document as given on its Title Page, then add an item

describing the Modified Version as stated in the previous sentence.

Preserve the network location, if any, given in the Document for public

access to a Transparent copy of the Document, and likewise the network

locations given in the Document for previous versions it was based on. These

may be placed in the "History" section. You may omit a network location for

a work that was published at least four years before the Document itself, or

if the original publisher of the version it refers to gives permission.

In any section entitled "Acknowledgements" or "Dedications," preserve the

section's title, and preserve in the section all the substance and tone of

each of the contributor acknowledgements and/or dedications given therein.

Preserve all the Invariant Sections of the Document,
unaltered in their text
and in their titles. Section numbers or the equivalent are not
considered
part of the section titles.
Delete any section entitled "Endorsements." Such a section
may not be
included in the Modified Version.
Do not retitle any existing section as "Endorsements" or to
conflict in
title with any Invariant Section.
If the Modified Version includes new front-matter sections or
appendices that
qualify as Secondary Sections and contain no material copied
from the
Document, you may at your option designate some or all of
these sections as
invariant. To do this, add their titles to the list of Invariant
Sections in
the Modified Version's license notice. These titles must be
distinct from any
other section titles.
You may add a section entitled "Endorsements," provided it
contains nothing
but endorsements of your Modified Version by various parties-
for example,
statements of peer review or that the text has been approved
by an
organization as the authoritative definition of a standard.
You may add a passage of up to five words as a Front-Cover
Text, and a passage
of up to 25 words as a Back-Cover Text, to the end of the list
of Cover Texts
in the Modified Version. Only one passage of Front-Cover Text
and one of

Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

COMBINING DOCUMENTS

You may combine the Document with other documents released under this License,

under the terms defined in section 4 above for modified versions, provided

that you include in the combination all of the Invariant Sections of all of

the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple

identical Invariant Sections may be replaced with a single copy. If there are

multiple Invariant Sections with the same name but different contents, make

the title of each such section unique by adding at the end of it, in

parentheses, the name of the original author or publisher of that section if

known, or else a unique number. Make the same adjustment to the section titles

in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the

various original documents, forming one section entitled "History"; likewise

combine any sections entitled "Acknowledgements," and any sections entitled

"Dedications." You must delete all sections entitled "Endorsements."

COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents

released under this License, and replace the individual copies of this License

in the various documents with a single copy that is included in the

collection, provided that you follow the rules of this License for verbatim

copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it

individually under this License, provided you insert a copy of this License

into the extracted document, and follow this License in all other respects

regarding verbatim copying of that document.

AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and

independent documents or works, in or on a volume of a storage or distribution

medium, does not as a whole count as a Modified Version of

the Document,

provided no compilation copyright is claimed for the compilation. Such a

compilation is called an "aggregate," and this License does not apply to the

other self-contained works thus compiled with the Document, on account of

their being thus compiled, if they are not themselves derivative works of the

Document.

If the Cover Text requirement of section 3 is applicable to these copies of

the Document, then if the Document is less than one quarter of the entire

aggregate, the Document's Cover Texts may be placed on covers that surround

only the Document within the aggregate. Otherwise they must appear on covers

around the whole aggregate.

TRANSLATION

Translation is considered a kind of modification, so you may distribute

translations of the Document under the terms of section 4.

Replacing Invariant

Sections with translations requires special permission from their copyright

holders, but you may include translations of some or all Invariant Sections in

addition to the original versions of these Invariant Sections.

You may include

a translation of this License provided that you also include the original

English version of this License. In case of a disagreement between the

translation and the original English version of this License,

the original

English version will prevail.

TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as

expressly provided for under this License. Any other attempt to copy, modify,

sublicense or distribute the Document is void, and will automatically

terminate your rights under this License. However, parties who have received

copies, or rights, from you under this License will not have their licenses

terminated so long as such parties remain in full compliance.

FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free

Documentation License from time to time. Such new versions will be similar in

spirit to the present version, but may differ in detail to address new

problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the

Document specifies that a particular numbered version of this License "or any

later version" applies to it, you have the option of following the terms and

conditions either of that specified version or of any later version that has

been published (not as a draft) by the Free Software Foundation. If the

Document does not specify a version number of this License, you may choose any

version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to Use This License for Your Documents

To use this License in a document you have written, include a copy of the

License in the document and put the following copyright and license notices

just after the title page:

Copyright (C) YEAR YOUR NAME. Permission is granted to copy, distribute

and/or modify this document under the terms of the GNU Free Documentation

License, Version 1.1 or any later version published by the Free Software

Foundation; with the Invariant Sections being LIST THEIR TITLES, with the

Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST. A

copy of the license is included in the section entitled "GNU Free

Documentation License".

If you have no Invariant Sections, write "with no Invariant Sections" instead

of saying which ones are invariant. If you have no Front-Cover Texts, write

"no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for

Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend

releasing these examples in parallel under your choice of free software

license, such as the GNU General Public License, to permit their use in free

software.

1 Tłumaczenie to jest dostępne w serwisie WWW polskiej sekcji Fundacji Wolnego Oprogramowania na stronie <http://gnu.org.pl/text/GFDL-pl.html>. Umieszczono tam również następujące zastrzeżenie prawne:

"To jest nieoficjalne tłumaczenie Licencji GNU Wolnej Dokumentacji na język polski. Nie zostało opublikowane przez Free Software Foundation i pod względem prawnym nie stanowi warunków rozpowszechniania tekstów stosujących GNU FDL - ustanawia je wyłącznie oryginalny angielski tekst licencji GNU FDL" - przyp. tłum.

2 Na temat autorstwa tłumaczenia podano następujące informacje:

1. Tłumaczenie na język polski: Krzysztof Łabanowski, w: Linux - podręcznik administratora sieci, Wydawnictwo RM, Warszawa 2000.

2. Poprawiono kilka literówek, zmieniono "wolnodostępny" na "wolny".

W.Kotwica ([wkotwica\(at\)post.pl](mailto:wkotwica(at)post.pl)).

3. Część omawiająca sposób stosowania GNU FDL do własnej dokumentacji została podesłana przez Przemka Sarnowskiego ([Przemyslaw.Sarnowski\(at\)nfosigw.gov.pl](mailto:Przemyslaw.Sarnowski(at)nfosigw.gov.pl))

- przyp. tłum.