

# Hakerzy – pierwsi artyści informatycy

„Tak, jestem przestępcą. Moim przestępstwem jest ciekawość”

*Loyd Blankenship (The Mentor)*

Jest początek lat 80. ubiegłego stulecia. Los rzucił mnie do Libii, ale nie trywialnie – nie pojechałem tam uczyć na uniwersytecie, aby zarobić parę złotych i kupić lub wypożyczyć mieszkanie w Polsce po powrocie. Powód był zgoła inny. Chodziło o trzymiesięczny kontrakt, jaki mój ówczesny pracodawca otrzymał na opracowanie i dostawę dyfraktometru i spektrometru neutronów dla powstającego libijskiego centrum badań jądrowych w miejscowości Tajura, niedaleko Trypolisu. Sprawa wagi niemal państwowej, stał za nią moskiewski Instytut Kurczatowa, który był głównym wykonawcą całości projektu, a pamiętajmy, że w tym Instytucie opracowano rosyjskie bomby atomowe. Z kolegą odpowiadaliśmy za opracowanie oprogramowania do wymienionych urządzeń pomiarowych.

## Niespodzianki

Nasze zdumienie nie miało granic, gdy – jeszcze w Polsce – zapoznaliśmy się z zadaniem. Oprogramowanie, z którym mieliśmy pracować, było całkowicie amerykańskie, firmy Data General, mimo że był to okres silnych napięć politycznych między Stanami Zjednoczonymi a Libią i obowiązywało absolutne embargo na transfer technologii amerykańskiej na ten rynek. Dla nas, jako fachowców, miało to o tyle znaczenie, że uzyskiwaliśmy bezpośredni dostęp do najnowszych technik programistycznych stosowanych w czołowych laboratoriach jądrowych na świecie. Zadanie było skomplikowane, ale mieliśmy wcześniejsze doświadczenie z oprogramowaniem konkurencyjnej firmy amerykańskiej Digital Equipment Corp. (DEC), poparte naszymi artykułami opublikowanymi na konferencjach międzynarodowych, więc znajomość rzeczy pozwoliła przejść przez sito kwalifikacyjne.

Na miejscu, w centrum libijskim, wszystko pod względem zawodowym szło gładko. Komputery nazywały się Nova



**Janusz Zalewski**

ukończył studia na Wydziale Elektroniki, doktorat obronił na Wydziale Elektrycznym Politechniki Warszawskiej. Po studiach pracował w Instytutach Badań Jądrowych w Warszawie i Świerku, komputeryzując eksperymenty z dziedziny fizyki i chemii jądrowej, a w 1989 r. wyjechał do USA, gdzie pracował w laboratoriach jądrowych oraz uczył informatyki na uczelniach w Teksasie i na Florydzie. Jest emerytowanym profesorem Florida Gulf Coast University i profesorem informatyki na Państwowej Uczelni Zawodowej im. Ignacego Mościckiego w Ciechanowie. Prywatnie zajmuje się tłumaczeniem na język polski literatury polskich Amerykanów oraz analizą twórczości literackiej amerykańskich bitników.

i były jednymi z pierwszych minikomputerów wyprodukowanych na świecie, a ich system operacyjny zwany RDOS (Real-time Disk Operating System) był pierwszym praktycznie działającym systemem operacyjnym czasu rzeczywistego, co miało ogromne znaczenie w sterowaniu urządzeniami. Nasz software użytkowy, napisany w zapomnianym już dzisiaj Fortranie, z którego wywoływane były procedury sterujące urządzeniami napisane w assemblerze, działał bez zarzutu, ale tylko do pewnego momentu.

Wykryliśmy wkrótce, że przy szybszych procesach zbierania danych z urządzeń (a chodziło o okresy czasowe rządu dziesiątek – a nawet czasem – pojedynczych milisekund) coś się psuje w naszym oprogramowaniu i ono po prostu kreskuje, zawieszając działanie całego systemu operacyjnego. Jak łątwo się domyśleć, spędzało nam to sen z powiek. Po pierwsze, na libijskiej pustyni trudno było o czyjąkolwiek fachową pomoc poza uzyskaniem jej od rosyjskich specjalistów, którzy przeszli trochę intensywniejsze szkolenie niż my, ale nie mieli aż tak dogłębnej wiedzy o RDOŚ-ie, a po drugie – nie mieliśmy dostępu do kodu źródłowego systemu operacyjnego, więc tak naprawdę nie można było niczego zdiagnozować.

Na szczęście, maszyna była wyposażona w profesjonalny debugger, w którym można było spokojnie prześledzić krokowe wykonywanie programu użytkowego (a więc „aplikacji”, używając dzisiejszego języka), nie tylko instrukcja po instrukcji w Fortranie, lecz także rozkaz po rozkazie w assemblerze i w języku maszynowym. Można było więc zatrzymać działanie programu w dowolnym momencie i sprawdzić stany rejestrów i danych w trakcie wykonywania rozkazu. Po paru nieprzespanych nocach doszukaliśmy się błędu w systemie operacyjnym, który przy odkładaniu wartości adresów na stos, w momentach wywoływania naszych procedur z Fortranu, błędnie obliczał jeden z adresów. Przy braku dostępu do kodu źródłowego poprawiliśmy ten błąd w dość prymitywny sposób, obchodząc operacje na stosie systemowym naszym własnym stosem, i wszystko zadziało jak trzeba.

## Pasowanie na hakera

Ale dlaczego przytaczam tę historyjkę? Kiedy napisałem raport z tego indycentu i z głupia frant wysłałem go do prezesa firmy Data General, Edsona de Castro, wkrótce otrzymałem teleks (dzisiaj mógłby to być faks lub mail), że w następnym poniedziałek ich główny inżynier chciałby przeprowadzić ze mną interview w filii firmy Data General we Frankfurcie w celu ewentualnego zatrudnienia. Okazało się, że z Warszawy nie mogę tam natychmiast pojechać, ale odbyłem rozmowę telefoniczną z Dennisem Damico, szefem działu Language Systems tej firmy, który powiedział: *you are a real hacker*.

Wtedy nie bardzo wiedziałem, co to słowo znaczy, ale niedługo potem wyjechałem na posiedzenie grupy roboczej „System Implementation Languages” IFIP-u i wysłuchałem wykładu Bjarne Stroustrupa (tak – samego twórcy języka C++), w który mówił o przeciążaniu operatorów<sup>1</sup> w C++ i w pewnym kontekście również użył słowa „hacker”. Na-

stępnego dnia kupiłem sobie książkę „The Hacker’s Handbook” (Londyn, 1985), żeby się o tym czegoś więcej dowiedzieć<sup>2</sup>. Od tego momentu zacząłem sam używać słowa hacker, głównie w znaczeniu pozytywnym.

Dziwnym zbiegiem okoliczności, w tym samym roku obejrzałem film „Gry wojenne” (reż. John Badham, 1983), wyświetlany także w Polsce, w którym młodzi adepci sztuki komputerowej podłączyli się zdalnie (wówczas było to możliwe tylko po liniach telefonicznych), aczkolwiek nieświadomie, do komputera w amerykańskiej bazie wojskowej NORAD i niemal doprowadzili do zmaterializowania fikcyjnych sytuacji z gier symulujących atak nuklearny, bo nikt z wojskowych nie przypuszczał że to, co widzą, jest zwykłą zabawą hakerów. Wtedy pomyślałem, że hacker to może być swego rodzaju włamywacz, z tym że używa klucza softwarowego, bo jego obiekty są nieuchwytnie fizycznie. Ale ta myśl nie zmieniła zasadniczego faktu, że hacker to ktoś, kto ma niebywałe pokłady energii twórczej.

Dla mnie nie był to jednak koniec, lecz dopiero początek rozważań o hakerstwie. Przy kolejnym wyjeździe na Zachód kolega, który wiedział o moich zainteresowaniach, aczkolwiek marginalnych w porównaniu z programowaniem systemów czasu rzeczywistego, podsunął mi króciutką notatkę zatytułowaną<sup>3</sup> „The Conscience of a Hacker”, która niedługo po rozpowszechnieniu otrzymała etykietkę *Manifestu Hakera*. Zawiera ona tylko 535 słów i jest do przeczytania w czasie krótszym niż sto sekund, ale jej doniosłość nie pozwala przygotowanemu czytelnikowi rozstać się z tematem kiedykolwiek.

” *Jak ktoś określił to wiele lat później, pisząc o tym manifestie na stronie Wikipedii<sup>4</sup>, „... jest coś takiego w hakerstwie, co wykracza poza egoistyczne pragnienie wykorzystania lub pokrzywdzenia innych osób” – nazywając to „kulturą hakera” lub też kamieniem węgielnym hakerstwa.*

Programistów i komputerowców zasługujących na określenie mianem hakera było wielu i jeszcze więcej ich będzie. Spośród najsłynniejszych można wymienić takie nazwiska, jak Kevin Mitnick i Kevin Poulsen, których działalność uznano za przestępczą i w konsekwencji otrzymali wyroki pozbawienia wolności, a także – Richard Greenblatt,

1 B. Stroustrup, Operator Overloading in C++. Proc. IFIP WG2.4 Conference on System Implementation Languages: Experience & Assessment, Canterbury, UK, September 1984.

2 H. Cornwall, The Hacker’s Handbook, Century Communications, 1985.

3 The Mentor (Lloyd Blankenship), The Conscience of a Hacker, 8 January 1986; URL: <http://phrack.org/issues/7/3.html>

4 Wikipedia, Hacker Manifesto. URL: [https://en.wikipedia.org/wiki/Hacker\\_Manifesto](https://en.wikipedia.org/wiki/Hacker_Manifesto)

Richard Stallman i Steve Wozniak, którzy reprezentują hakerów zainteresowanych zgłębianiem tajników technologii komputerowych w celu udostępniania ich jak najszerszym gremiom. Na te tematy powstają grube tomy, ale w tym krótkim artykule, aby zgłębić, czy tkwi w hakerstwie jakiś arcyzm, chcę się zatrzymać na kimś, kto nie jest ani specjalnie bardzo znany, ani popularny – to Jamie Zawinski.

## Wzorzec

Wstyd się przyznać, ale o Zawinskim nie dowiedziałem się ani na żadnej konferencji komputerowej, ani od profesjonalnych informatyków, ale od jego mamy, poetki amerykańskiej polskiego pochodzenia, pani Andreny Zawinski<sup>5</sup>, którą spotykałem czasami na imprezach literackich w San Francisco i okolicy. Dla tych, którzy – podobnie jak ja – nie słyszeli dostatecznie wcześniej o Zawinskim, wystarczy powiedzieć, że był on twórcą pierwszej wersji przeglądarki Netscape dla Unixa, a potem współtworzył fundację Mozilla, znaną z rozpowszechniania nieodpłatnego oprogramowania. Nie to jest jednak dla nas ważne w kontekście tego artykułu, lecz rozwój jego kariery.

Chcę się przyjrzeć bliżej postaci Zawinskiego, aby na tym przykładzie zidentyfikować kilka cech, które charakteryzują twórcze postawy i umiejętności hakerów. Trudno to zweryfikować, ale na pierwszym miejscu wymienilibym geny. Choć zabrzmiało to trywialnie, matka jako poetka z pewnością musiała przekazać synowi jakiś materiał genetyczny predystynujący do działań twórczych. Drugi równie ważny czynnik, jeśli nie ważniejszy, to otoczenie. Jak Jamie wspomina w swoim wywiadzie do książki „Sztuka kodowania”<sup>6</sup>, lata młodości licealnej spędził w Pittsburghu, gdzie jako piętnastolatek zaimponował akademikom na uniwersytecie Carnegie Mellon (CMU) znajomością tajników Lispu i otrzymał *carte blanche* na dostęp do laboratorium, gdzie pracowano nad sztuczną inteligencją, i w którym go później zatrudniono. Uważam, że atmosfera współzawodnicstwa a jednocześnie współpracy jest tam niepowtarzalna. Tak sądziłem o wielu kampusach uniwersyteckich w Stanach, ale spostrzeżenie to w pełni potwierdziło się, gdy kiedyś odwiedziwszy CMU, wracałem wieczorem z jakiegoś nudnego konferencyjnego przyjęcia i kolega zaciągnął mnie do kawiarenki uniwersyteckiej otwartej do godziny drugiej na ranem! Z taką otwartością na kampusie nie spotkałem się nigdzie indziej.

Do tych charakterystyk, które są w znacznym stopniu niezależne od kandydata na hakera, dochodzą jego cechy immanentne. Pierwszą z nich jest niezwykle silna wewnętrzna motywacja, jakiś stanowczy, niezbywalny napęd, niepozwalający na oderwanie się od poszukiwania rozwiązania aż do skutku, bez względu na konsekwencje. Zgłębianie się w ko-

dzie, szukanie najprzeróżniejszych narzędzi i sztuczek do rozwiązania problemu, operowanie nie tylko deterministycznymi bitami, lecz też wymyślnymi heurystykami, aby obejść rzeczywiste i potencjalne przeszkody stojące na drodze do programistycznego celu. Byłem bardzo dumny z siebie, że na libijskiej pustyni za pomocą debuggera zhakowałem amerykański system operacyjny. Ale w podobnej sytuacji Zawinski poszedł o rząd wielkości dalej, bo nie mogąc sobie poradzić z prawie identycznym problemem (jego program w Lispie kreszował przypadkowo po wykonaniu kilkuset instrukcji), doszedł do przekonania, że to musi być błąd w samym debuggerze i poprawił go, o czym z wykrzyknikiem mówi w swoim wywiadzie: „There was a bug in GDB!” w „Sztuka kodowania”.

Pasja hakerska znajduje ujście nie tylko w wyszukiwaniu metod dotarcia do celu, ale też w udostępnianiu tych osiągnięć szerszemu ogółowi. No bo cóż więcej taki haker może zrobić, jeśli nie robi tego dla pieniędzy? Jeśli już zademonstrował swoje umiejętności i osiągnął wewnętrzną satysfakcję, to może tylko oddać swoje dzieło do dyspozycji innym. Może to być jakiś rodzaj altruizmu ukierunkowanego na cele szersze od początkowo planowanych. W tym aspekcie Zawinski zachował się jak prawdziwy haker, gdy namówił – jak głosi fama – firmę Netscape, aby upubliczniła w 1998 r. kod źródłowy swej przeglądarki.



Czy te cechy osobowościowe wystarczają, aby hakerów nazywać artystami? Można na to spojrzeć w sposób bardziej uczony, na przykład dowodząc, że problemy etyczne łączące często z tą dziedziną są w istocie problemami estetycznymi, a więc nawet szerszymi niż artystyczne<sup>7</sup>. Ale gdy sam się temu przyglądam, przychodzi mi na myśl pewna nieodłączna cecha artystów – tworzą sztukę dla sztuki! I to może być ostatecznym wyróżnikiem hakera jako artysty. Tak więc, czy haker lub szerzej – informatyk – jest artystą? Nie ulega wątpliwości, że na pewno jest twórcą, ale żeby to dokładniej skwantyfikować, należałoby całe zjawisko zbadać głębiej.

Autor dziękuje p. Bartoszowi Klonowskiemu z Fundacji SLOW (<https://beslow.pl/>) za inspirację do podjęcia tego tematu.



W kawiarence uniwersyteckiej w Carnegie Mellon można było też wypić piwo, aczkolwiek bardzo nędzne.

<sup>5</sup> Polish American Writers – Andrena Zawinski. URL: <http://pol-am-writers.org/>

<sup>6</sup> P. Seibel, Sztuka kodowania. Sekrety wielkich programistów. Helion, 2011.

<sup>7</sup> M. Ommeln, L. Pimenidis, The Art of Hacking, KIT Forschungsbericht No. 83, Karlsruher Institut für Technologie, 2018.