Brian W. Kernighan

# JAK UNIX TWORZYŁ HISTORIĘ


Helion

Tytuł oryginału: UNIX: A History and a Memoir

Tłumaczenie: Piotr Cieślak

ISBN: 978-83-283-7163-7

Brian W. Kernighan: UNIX: A History and a Memoir, 1st Edition



Helion S.A.
ul. Kościuszki 1c, 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/jakuth*
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

# SPIS TREŚCI

# 1

# BELL LABS

„Jedna polityka, jeden system, powszechność usług”

— misja firmy AT&T w brzmieniu z 1907 roku

„Na pierwszy rzut oka, gdy natrafi się na nią w zaskakująco wiejskiej scenerii, główna siedziba Bell Telephone Laboratories w New Jersey wygląda jak duża i nowoczesna fabryka, którą w pewnym sensie istotnie jest. Ale to fabryka pomysłów, więc jej linie produkcyjne są niewidoczne”.

— Arthur Clarke, *Voice Across the Sea* (1974); cytat w brzmieniu zamieszczonym w książce *The Idea Factory* Jona Gertnera (2012)

Aby zrozumieć, jak doszło do powstania Uniksa, należy zapoznać się z Bell Labs, a zwłaszcza z działaniem tego ośrodka i kreatywnym środowiskiem, które w owym czasie w nim istniało.

AT&T, American Telephone and Telegraph Company, powstała z połączenia wielu lokalnych firm telekomunikacyjnych z całych Stanów Zjednoczonych. Już na początku istnienia firmy szefowie AT&T zdali sobie sprawę z potrzeby utworzenia jednostki badawczej, która na bieżąco zajmowałaby się problemami naukowymi i inżynieryjnymi, na jakie napotykała firma będąca narodowym operatorem telekomunikacyjnym. I tak w 1925 roku założono filię badawczo-rozwojową Bell Telephone Laboratories, która miała zajmować się tymi zagadnieniami. Choć pełna nazwa ośrodka była regularnie skracana do „Bell Labs”, „BTL”, czy nawet po prostu „the Labs”, w centrum zainteresowania od początku była telefonia.

Oddział Bell Labs pierwotnie mieścił się przy 463 West Street w Nowym Jorku, ale u progu II wojny światowej większą część operacji przeniesiono poza miasto. Firma AT&T była mocno zaangażowana w działania wojenne i służyła doświadczeniem w rozmaitych ważnych militarnie obszarach — oczywiście przede wszystkim chodziło o systemy łączności, ale także o urządzenia obliczeniowe sterujące ogniem dział przeciwlotniczych, radary i kryptografię. Część owych prac prowadzono na przedmieściach i w wiejskich rejonach New Jersey, 33 kilometry na zachód od Nowego Jorku. Największa placówka mieściła się w obszarze zwanym Murray Hill, wchodzącym w skład miasteczek New Providence i Berkeley Heights.

Rysunek 1.1 pozwala zorientować się w tej okolicy; 463 West Street znajduje się nad rzeką Hudson, nieopodal znacznika autostrady 9A. Bell Labs w Murray Hill zostało ulokowane na granicy między New Providence a Berkeley Heights, na północ od autostrady międzystanowej 78. Obie lokalizacje zostały zaznaczone na mapie kropkami.

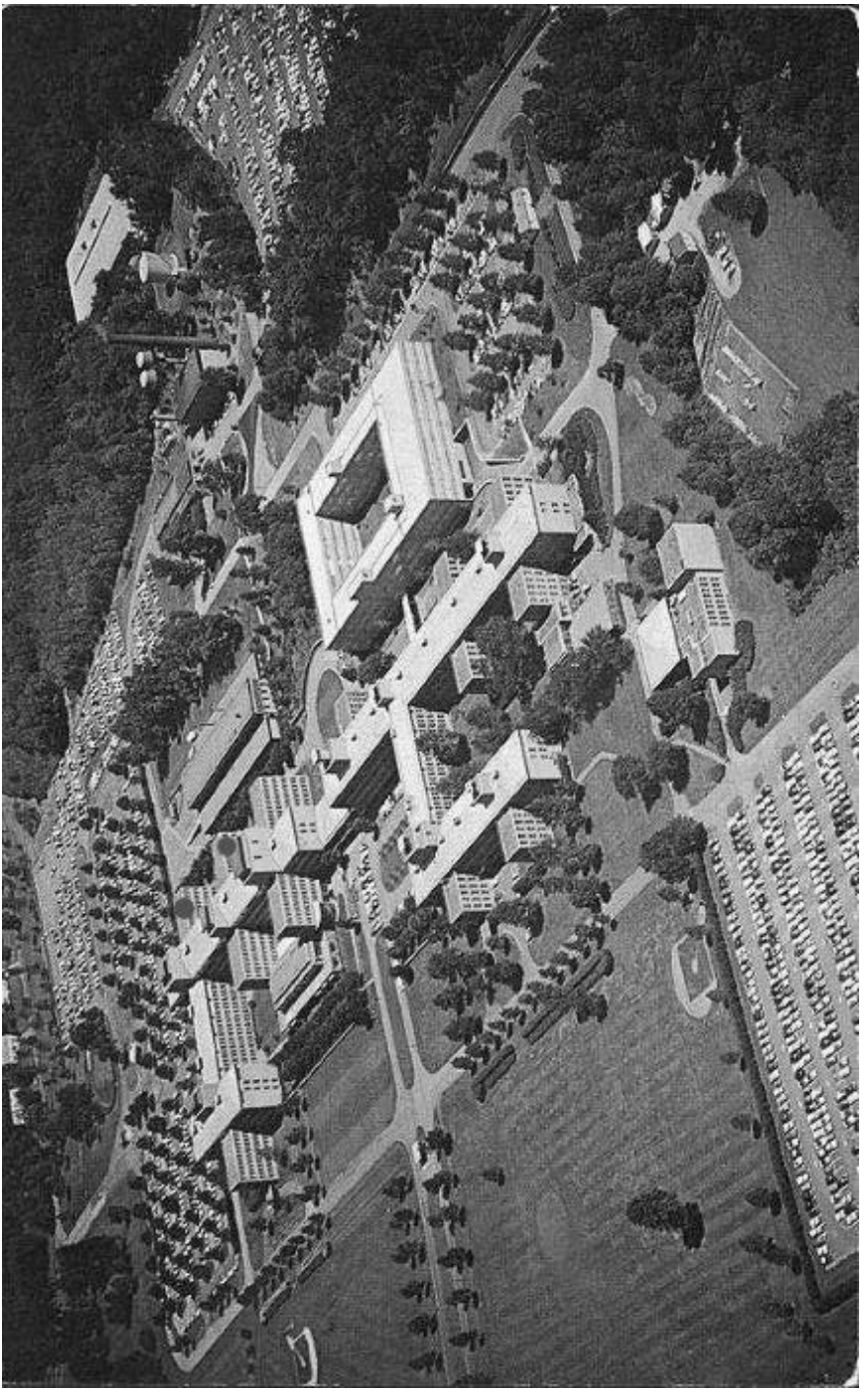
Do Murray Hill stopniowo przenosiło się coraz więcej przedsięwzięć, aż wreszcie w 1966 roku biura Bell Labs na West Street zupełnie opustoszały. W latach 60. w Murray Hill pracowało ponad trzy tysiące osób, spośród których co najmniej tysiąc miało stopień doktora w dziedzinach takich jak fizyka, chemia, matematyka i różne gałęzie inżynierii.

Rysunek 1.2 przedstawia zdjęcie lotnicze kompleksu Murray Hill w 1961 roku. W jego skład wchodziły trzy główne budynki. Budynek 1 znajduje się w lewej dolnej części zdjęcia, budynek 2 w lewej górnej, a budynek 3 to ten z otwartym dziedzińcem pośrodku. Aż do lat 70., kiedy to wzniesiono dwa nowe biurowce, od końca budynku 1 do przeciwległego końca budynku 2 prowadził długi, czterystumetrowy korytarz.

Od stażu w 1967 roku do przejścia na emeryturę w roku 2000, spędziłem w budynku 2 ponad trzydzieści lat. Moje gabinety znajdowały się w bocznych skrzydłach oznaczonych kropkami, na czwartym (ostatnim) piętrze. Klatka schodowa numer 9 znajdowała się na samym końcu budynku 2, a klatka numer 8 o jedno skrzydło bliżej środka. Przez większość wczesnych lat rozwoju systemu, pokój uniksowy znajdował się na poddaszu między klatkami schodowymi 8 i 9.

Rysunek 1.3 przedstawia satelitarne zdjęcie Bell Labs wykonane na potrzeby map Google w 2019 roku. Budynek 6 (w lewej dolnej części, oznaczony pinezką) oraz budynek 7 (w prawej górnej części) zostały dobudowane na początku lat 70., a przez kilka lat począwszy od 1996 roku pierwszy z nich był siedzibą firmy


**Rysunek 1.1.** *Od Nowego Jorku do Murray Hill w New Jersey*



**Rysunek 1.2.** *Bell Labs w 1961 roku. (Dzięki uprzejmości Bell Labs)*


**Rysunek 1.3.** *Bell Labs w 2019 roku; budynek 6 znajduje się w lewej dolnej części*

Lucent Technologies. To niezwykłe, jak wielka część historii firmy zawiera się w etykietach wyświetlanych na mapach Google: „Bell Labs" przy znaczniku (pinezce), „Lucent Bell Labs" na drodze wyjazdowej, „Alcatel-Lucent Bell Labs" na drodze wjazdowej i „Nokia Bell Labs" przy szczytowych pomieszczeniach dla kadry menedżerskiej w budynku 6.

Nie czuję się na tyle dobrze zorientowany w historii Bell Labs, by ją tutaj szczegółowo przedstawić, lecz na szczęście zostało to już wyśmienicie zrobione przez innych pisarzy. Szczególnie podobała mi się książka *The Idea Factory* (Fabryka pomysłów) Jona Gertnera, która koncentruje się na naukach fizycznych, oraz *The Information* (Informacja) Jamesa Gleicka, poświęcona kwestiom informatycznym. Niezwykle obszerna — blisko pięć tysięcy stron w siedmiu tomach — oficjalna publikacja Bell Labs, zatytułowana *A History of Science and Engineering in the Bell System* (Historia nauki i inżynierii w Bell System), jest rzetelna i wiarygodna, a we wszystkich fragmentach, z którymi się zapoznałem, także ciekawa.

## 1.1. NAUKI FIZYCZNE W BELL LABS

We wczesnych latach istnienia Bell Labs większość prowadzonych w ośrodku badań dotyczyła fizyki, chemii, materiałów i systemów komunikacyjnych. Naukowcy mieli wyjątkową swobodę w realizacji własnych zainteresowań, lecz otoczenie oferowało takie bogactwo praktycznych problemów, że nietrudno było zająć się badaniami, które byłyby zarówno ciekawe z naukowego punktu widzenia, jak i przydatne dla firmy Bell System i całego świata.

W Bell Labs dokonano bardzo wielu naukowych i technicznych przełomów, które odmieniły świat. Najdonioślejszym spośród nich było wynalezienie w 1947 roku tranzystora przez Johna Bardeena, Waltera Brattaina i Williama Shockleya, którzy starali się ulepszyć wzmacniacze do obwodów telefonii dalekosiężnej. Tranzystor był owocem fundamentalnych badań nad właściwościami materiałów półprzewodnikowych, które miały na celu opracowanie urządzeń bardziej wytrzymałych i mniej energochłonnych niż lampy próżniowe, będące w latach 40. ubiegłego wieku podstawowym podzespołem sprzętu telekomunikacyjnego oraz pierwszych komputerów.

Wynalezienie tranzystora zostało w 1956 roku uhonorowane Nagrodą Nobla w dziedzinie fizyki — jednym z dziewięciu Nobli przyznanych naukowcom za inicjatywy, które przynajmniej w pewnym stopniu powstały w Bell Labs. Wśród innych ważnych wynalazków należy wymienić wzmacniacz ujemnego sprzężenia zwrotnego, ogniwo słoneczne, laser, telefonię komórkową, satelitę telekomunikacyjnego i światłoczuły element CCD (dzięki któremu działa na przykład aparat fotograficzny w smartfonie).

Z grubsza rzecz ujmując, od lat 60. do lat 80. ubiegłego stulecia w dziale badawczo-rozwojowym w Bell Labs (głównie w Murray Hill) pracowało 3000 osób, a kolejnych 15 – 25 tysięcy, rozlokowanych w wielu różnych miejscach, zajmowało się projektowaniem sprzętu i systemów dla Bell System, często korzystając z owoców pracy tego działu. To ogromny sztab ludzi. Kto im wszystkim płacił?

Firma AT&T była de facto monopolistą, ponieważ świadczyła usługi telefoniczne dla większości mieszkańców Stanów Zjednoczonych, lecz jej zdolność do czerpania profitów z tego monopolu była ograniczona. Przedsiębiorstwo podlegało organom federalnym i stanowym, kontrolującym ceny rozmaitych usług, i nie mogło brać udziału w przedsięwzięciach, które nie miały bezpośredniego związku z operacjami telekomunikacyjnymi.

Ten reżim regulacyjny sprawdzał się przez wiele lat. Firma AT&T miała dostarczać usługi wszystkim („powszechność usług"), bez względu na dystans i opłacalność. Rekompensatą były stabilne i przewidywalne przychody.

W ramach tego porozumienia AT&T przeznaczała niewielką część dochodu na Bell Labs, z myślą o ciągłym podwyższaniu jakości usług telekomunikacyjnych. Innymi słowy, Bell Labs był finansowany z czegoś na kształt bardzo niewielkiego podatku od każdej rozmowy telefonicznej w kraju. Według artykułu A. Michaela Nolla, AT&T przeznaczała około 2,8 procenta swoich przychodów na badania i rozwój, a około 0,3 procenta na badania podstawowe[1]. Nie jestem pewny, na ile dobrze sprawdziłby się taki model dzisiaj, lecz przez dziesięciolecia pozwalał on na nieustanne ulepszanie systemu telekomunikacyjnego i doprowadził do wielu fundamentalnych odkryć naukowych.

Stabilne finansowanie było kardynalnym warunkiem powodzenia badań. Oznaczało, że AT&T mogła przyjąć długofalową perspektywę, a naukowcy z Bell Labs mieli swobodę w eksplorowaniu obszarów, które mogły nie przynieść krótkoterminowych — albo nawet żadnych — korzyści. To zupełnie inaczej niż w dzisiejszym świecie, w którym wszelkie plany sięgają zaledwie kilka miesięcy naprzód, a wiele uwagi poświęca się spekulowaniu na temat wyników finansowych w następnym kwartale.

[1] Prace eksperymentalne lub teoretyczne podejmowane przede wszystkim w celu zdobycia nowej wiedzy o podstawach zjawisk i obserwowalnych faktów, bez nastawienia na bezpośrednie zastosowanie komercyjne (Wikipedia) — *przyp. tłum.*

## 1.2. TELEKOMUNIKACJA I INFORMATYKA

Ośrodek Bell Labs był oczywiście pionierem w projektowaniu, konstruowaniu i ulepszaniu systemów komunikacyjnych, przy czym termin ten obejmował wszystko — od opracowywania sprzętu konsumenckiego, takiego jak aparaty telefoniczne, aż po infrastrukturę central, wież transmisyjnych czy przewodów światłowodowych.

Ten szeroki zakres praktycznych zagadnień prowadził niekiedy do postępów w naukach podstawowych. Na przykład w 1964 roku Arno Penzias i Robert Wilson próbowali ustalić, co powoduje niepożądany szum w antenie, której w Bell Labs używano do odbierania sygnałów radiowych odbijających się od balonów projektu Echo[2]. Doszli do wniosku, że szum ten pochodzi z promieniowania tła, będącego pozostałością po kosmicznym Wielkim Wybuchu na początku Wszechświata. Odkrycie to przyniosło w 1978 roku Penziasowi i Wilsonowi Nagrodę Nobla w dziedzinie fizyki. (Arno powiedział: „Większość ludzi dostaje Nobla za rzeczy, które chcą znaleźć. My dostaliśmy go za coś, czego chcieliśmy się pozbyć").

Innym aspektem misji Bell Labs było rozwijanie narzędzi matematycznych, pozwalających na lepsze zrozumienie działania systemów komunikacyjnych. Najważniejszym rezultatem tych badań było stworzenie przez Claude'a Shannona teorii informacji, która po części wyrosła na gruncie jego studiów nad kryptografią w czasie II wojny światowej. W 1948 roku w „Bell System Technical Journal" ukazał się jego artykuł *A Mathematical Theory of Communication*, w którym Shannon wyjaśnił podstawowe cechy i ograniczenia dotyczące ilości informacji, jaką da się przesłać przez system komunikacyjny. Shannon pracował w Murray Hill od wczesnych lat 40. do 1956 roku, po czym wrócił na Massachusetts Institute of Technology (MIT), którego był absolwentem. Zmarł w 2001 roku, w wieku 84 lat.

W miarę jak komputery stawały się coraz potężniejsze i tańsze, rosły ich zastosowania — zaczęto używać ich do coraz większej liczby analiz danych oraz rozbudowanego modelowania i symulowania fizycznych systemów i procesów. Firma Bell Labs zajmowała się komputerami i informatyką od lat 30. ubiegłego wieku, a do późnych lat 50. dysponowała ośrodkiem obliczeniowym z dużymi komputerami centralnymi.

[2] Pionierski eksperyment w dziedzinie komunikacji satelitarnej, polegający na umieszczeniu na orbicie balonu o średnicy około 30 metrów i metalizowanej powłoce, która miała odbijać sygnały radiowe. Eksperymenty z kilkoma balonami tego rodzaju prowadzono w latach 60. ubiegłego wieku — *przyp. tłum.*

Grupę badawczą zajmującą się informatyką utworzono na początku lat 60. Trafiły do niej osoby zajmujące się wcześniej badaniami matematycznymi oraz część pracowników obsługujących duży centralny komputer w Murray Hill. Powstały konglomerat nazwano Computing Science Research Center, a choć przez krótki czas nadal wykonywał on zadania komputerowe dla całego ośrodka w Murray Hill, nie pełnił funkcji usługowej, lecz stanowił część działu badawczego. W 1970 roku grupa, która zarządzała zapleczem komputerowym, została przeniesiona do odrębnej organizacji.

## 1.3. BWK W BTL[3]

W tej części rozdziału zawarłem sporo osobistych wspomnień, które — mam nadzieję — dadzą Ci pewne wyobrażenie o szczęśliwych zbiegach okoliczności, w wyniku których zająłem się zawodowo informatyką i trafiłem do Bell Labs, miejsca, gdzie jak nigdzie indziej dało się zgłębiać tę dziedzinę.

Urodziłem się w Toronto i studiowałem na tamtejszym uniwersytecie. Wybrałem kierunek o nazwie fizyka inżynieryjna (przemianowany potem na nauki inżynieryjne), przeznaczony dla wszystkich tych, którzy nie wiedzieli jeszcze, na czym konkretnie chcą się skupić. Studia ukończyłem w roku 1964, czyli w czasach, w których dzisiejsza informatyka stawiała pierwsze kroki. Po raz pierwszy w życiu zobaczyłem komputer na trzecim roku studiów. Na całej uczelni była tylko jedna taka duża maszyna, IBM 7094, w owym czasie będąca szczytem możliwości technicznych. Komputer dysponował ferrytową pamięcią podstawową o pojemności 32K (32 768) słów w architekturze 36-bitowej (dziś powiedzielibyśmy, że miał jej 128 kilobajtów) oraz pewną ilością pamięci dodatkowej w postaci wielkich, mechanicznych dysków twardych. Kosztował trzy miliony ówczesnych dolarów amerykańskich i znajdował się w dużym, klimatyzowanym pomieszczeniu, obsługiwanym przez wykwalifikowanych operatorów; zwykli ludzie — a zwłaszcza studenci — nie mieli do niego dostępu.

W rezultacie na studiach nie miałem wiele do czynienia z komputerami, choć próbowałem nauczyć się języka programowania Fortran. Dobrze rozumiem każdego, kto miał problemy z napisaniem swojego pierwszego programu. Dysponowałem znakomitą książką Daniela McCrackena o Fortranie II i rozumiałem zasady, lecz

[3] BWK to inicjały autora (Brian W. Kernighan), BTL jest akronimem nazwy firmy Bell Telephone Laboratories — *przyp. tłum.*

nie potrafiłem niczego stworzyć — odbijałem się od bariery koncepcyjnej, którą zdaje się napotykać wielu ludzi.

W wakacje przed ostatnim rokiem studiów dostałem pracę w Imperial Oil w Toronto, firmie, która zajmowała się tworzeniem oprogramowania optymalizującego pracę rafinerii. (Imperial Oil była częściowo własnością korporacji Standard Oil z New Jersey, która w 1972 roku została przemianowana na Exxon).

Z perspektywy czasu uważam, że nie byłem dobrym stażystą. Strawiłem całe lato na próbach napisania w Cobolu wielkiego programu do analizowania danych z rafinerii. Nie pamiętam już, na czym konkretnie miało polegać jego działanie, za to świetnie pamiętam, że nigdy nie zadziałał. Tak naprawdę nie umiałem programować, Cobol nie słynął ze wsparcia dla dobrej organizacji kodu, a programowania strukturalnego jeszcze nie wynaleziono, mój program był więc niekończącą się sekwencją instrukcji `IF`, które rozgałęziały się w celu wykonania jakiejś operacji, gdy już udało mi się zorientować, na czym owa operacja ma polegać.

Oprócz tego, na komputerze IBM 7010 firmy Imperial Oil próbowałem uruchamiać programy w Fortranie, bo trochę go znałem — na pewno lepiej niż Cobola — a język ten bardziej nadawał się do analizy danych. O tym, że na IBM 7010 nie było kompilatora Fortrana, dowiedziałem się dopiero po kilku tygodniach walki z Job Control Language (JCL), bo komunikaty błędów JCL były tak enigmatyczne, że nikt nie zdawał sobie z tego sprawy.

Wróciwszy na ostatni rok studiów po tym nieco frustrującym lecie nadal byłem bardzo zainteresowany informatyką. Nie było żadnych oficjalnych kursów informatycznych, napisałem więc pracę magisterską na temat sztucznej inteligencji, która była wówczas gorącym tematem. Dowodzenie twierdzeń, programy do gry w szachy i warcaby oraz narzędzia do automatycznego tłumaczenia języków naturalnych — wszystko to wydawało się być w zasięgu ręki, tylko wymagało odrobiny programowania.

Po ukończeniu studiów w 1964 roku nie miałem pojęcia, co robić dalej i tak jak wielu absolwentów odłożyłem tę decyzję na później, idąc na studia podyplomowe. Złożyłem podania do sześciu amerykańskich uczelni (co w ówczesnej Kanadzie było rzadkością) i traf chciał, że zostałem zaaprobowany przez więcej niż jedną — w tym MIT i Princeton. W tej drugiej dowiedziałem się, że zrobienie doktoratu zajmuje przeciętnie trzy lata, w pierwszej zaś, że aż siedem. Uczelnia w Princeton zaoferowała pełne stypendium, tymczasem w MIT miałem przez 30 godzin tygodniowo pracować jako asystent naukowy. Decyzja wydawała się oczywista, a ponieważ mój bliski przyjaciel Al Aho, który ukończył Uniwersytet

w Toronto rok wcześniej, także wybrał Princeton, poszedłem w jego ślady. Okazało się, że był to bardzo trafny wybór.

W 1966 roku znów mi się poszczęściło — dostałem letni staż na MIT, między innymi dzięki innemu absolwentowi Princeton, Lee Varianowi, który w 1965 zrobił tam kawał dobrej roboty. Te wakacje spędziłem na poznawaniu Compatible Time-Sharing System (CTSS) oraz pisaniu programów w MAD (Michigan Algorithm Decoder; to jeden z wariantów języka ALGOL 58) na potrzeby narzędzi dla nowego systemu operacyjnego o nazwie Multics, o którym będzie mowa w rozdziale 2. (Nazwę Multics początkowo zapisywano jako MULTICS, lecz zapis małymi literami jest przyjemniejszy dla oczu, podobnie jak w przypadku nazwy UNIX i innych terminów podawanych wersalikami. Choć oznacza to odejście od historycznej dokładności, będę się posługiwał ładniej wyglądającymi nazwami).

Moim oficjalnym przełożonym w MIT był profesor Fernando Corbató, nazywany przez wszystkich „Corbym", wspaniały dżentelmen, twórca CTSS i szef projektu Multics. W 1990 roku Corby zdobył Nagrodę Turinga za położenie fundamentów pod rozwój systemów z podziałem czasu (ang. *time-sharing systems*). Zmarł w lipcu 2019 roku w wieku 93 lat.

Oprócz tego, że kierował projektowaniem i wdrażaniem CTSS i Multicsa, Corby był pomysłodawcą zastosowania haseł dostępu do komputera. W przypadku przetwarzania wsadowego właściwie nie było takiej potrzeby, lecz na komputerze wyposażonym w system podziału czasu i współdzielony system plików należało wdrożyć jakieś mechanizmy bezpieczeństwa.

Wciąż pamiętam oba krótkie, łatwe do wymówienia hasła, które zostały dla mnie automatycznie wygenerowane, gdy gościłem na MIT latem 1966 roku. Potrzebne były dwa hasła, a nie jedno, bo wskutek pewnego nieporozumienia, któregoś dnia plik z hasłami został zamieniony z plikiem zawierającym wiadomość dnia, wyświetlaną po zalogowaniu. Ponieważ hasła były przechowywane w niezaszyfrowanej postaci, każdy, kto zalogował się do systemu, ujrzał hasła wszystkich pozostałych.

Moje biuro — o ile mnie pamięć nie myli, z nikim go nie dzieliłem — należało do profesora Josepha Weizenbauma, który wyjechał na letni urlop. Na początku 1966 roku Weizenbaum opublikował opis Elizy, programu symulującego dialog z psychologiem rogersowskim. Eliza była jedną z pierwszych prób wykorzystania języka naturalnego do interakcji z komputerem. Program był zadziwiająco dobry, przynajmniej przy pierwszym zetknięciu, i lubiłem pokazywać go odwiedzającym.

Dziś w internecie jest mnóstwo symulatorów Eliza i nadal sprawiają one frajdę przez kilka pierwszych chwil, choć słabości algorytmu szybko wychodzą na jaw.

```
ELIZA: Powiedz proszę, co cię martwi.
JA: Próbuję napisać książkę o Uniksie.
ELIZA: Przyszedłeś do mnie, bo próbujesz napisać książkę
o Uniksie?
JA: Tak.
ELIZA: Brzmisz pewnie.
JA: Tak.
ELIZA: Wydajesz się pozytywnie nastawiony.
JA: Tak.
ELIZA: Rozumiem.
```

Latem 1967 roku kolejny raz spotkało mnie wyjątkowe szczęście: dostałem staż w Bell Labs w Murray Hill, a konkretnie w Computing Science Research Center pod kierunkiem Douga McIlroya (rysunek 1.4). Doug zasugerował, żebym zbadał jakiś problem dotyczący alokowania pamięci — było to jedno z zagadnień leżących w obszarze jego długofalowych zainteresowań. Jak na stażystę przystało, zajmowałem się wszystkim, tylko nie tym, i ostatecznie zrobiłem coś zupełnie innego: opracowałem bibliotekę funkcji, które ułatwiały przetwarzanie list w programach napisanych w Fortranie. Lato spędziłem na pisaniu zwięzłego języka asemblera dla ówczesnego dużego komputera w Murray Hill, GE 635. Z technicznego punktu widzenia był to czystszy i bardziej uporządkowany IBM 7094, będący zarazem prostszym wariantem modelu GE 645, zaprojektowanego specjalnie dla Multicsa. Był to ostatni raz, gdy pisałem język asemblera, ale chociaż moje podejście było zasadniczo błędne, pracowało mi się świetnie i programowanie wciągnęło mnie bez reszty.

**Rysunek 1.4.** *Doug McIlroy, ok. 1981 roku. (Dzięki uprzejmości Gerarda Holzmanna)*

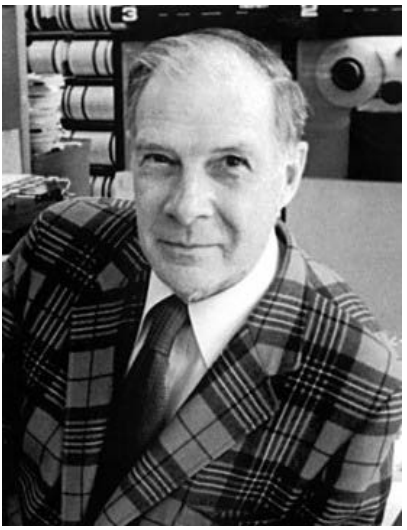
## 1.4. PRZESTRZEŃ BIUROWA

Czasami wszystko zależy od miejsca.

Moje biuro jako stażysty w 1967 roku znajdowało się na czwartym piętrze budynku 2, na korytarzu przy klatce schodowej 8. Pierwszego dnia pracy siedziałem u siebie (ach, te stare, dobre czasy, kiedy nawet stażysta mógł cieszyć się luksusem własnego gabinetu) i zastanawiałem się, co robić, gdy nagle około 11.00 w progu stanął starszy facet i powiedział: „Cześć, jestem Dick [niezrozumiały]. Chodźmy na lunch".

Pomyślałem, że czemu nie. Z lunchu nie zapamiętałem nic, pamiętam za to, że później, gdy Dick [niezrozumiały] dokądś sobie poszedł, ja ukradkiem przemknąłem korytarzem pod drzwi jego gabinetu, by przeczytać tabliczkę z nazwiskiem. Richard Hamming! Mój sąsiad zza ściany był słynnym wynalazcą kodów korygujących błędy i autorem podręcznika do kursu analizy numerycznej, który dopiero co ukończyłem.

Zaprzyjaźniłem się z Dickiem (rysunek 1.5). Miał swoje zdanie i nie bał się go wyrażać, co — jak przypuszczam — zniechęcało do niego niektórych ludzi, lecz ja dobrze się czułem w jego towarzystwie, a rady, których udzielał mi przez lata, zawsze procentowały.



**Rysunek 1.5.** *Dick Hamming, ok. 1975 roku, w typowej dla siebie marynarce w kratę. (Wikipedia)*

Był kierownikiem działu, ale w jego dziale nie było ludzi, co wydało mi się dość dziwne. Powiedział mi, że ciężko pracował, by osiągnąć to szczególne połączenie odpowiedniego tytułu z brakiem odpowiedzialności, co doceniłem znacznie później, gdy sam zostałem kierownikiem kilkunastoosobowego działu.

Byłem tam latem 1968 roku, gdy dowiedział się, że otrzymał Nagrodę Turinga, uważaną dziś za odpowiednik Nagrody Nobla w dziedzinie informatyki. Zareagował z właściwą sobie ironią: powiedział, że skoro Nagroda Nobla była wtedy warta sto tysięcy dolarów, a Nagroda Turinga dwa tysiące, zdobył 2 procent Nagrody Nobla. (To była trzecia w historii Nagroda Turinga; pierwsze dwie trafiły do Alana Perlisa i Maurice'a Wilkesa, innych pionierów informatyki). Dick został wyróżniony za swoje prace nad metodami numerycznymi, automatycznymi systemami kodowania i kodami wykrywającymi i korygującymi błędy.

To dzięki Dickowi zacząłem pisać książki, co wyszło mi na dobre. Miał dość marną opinię o programistach, którzy jego zdaniem byli wyszkoleni źle albo wcale. Wciąż brzmią mi w uszach jego słowa:

> *„Dajemy im słownik, reguły gramatyczne i mówimy: »Teraz jesteś świetnym programistą, młody«".*

Uważał, że programowania trzeba nauczać tak, jak uczy się pisania. Powinny istnieć zasady stylistyczne, które pozwalają odróżniać zły kod od dobrego, a programistom należy kłaść do głowy, by pisali poprawnie stylistycznie i doceniali dobry styl kodowania.

Nie zgadzaliśmy się wprawdzie co do sposobu realizacji tego celu, lecz jego pomysł był bardzo słuszny i skłonił mnie do napisania pierwszej książki, *The Elements of Programming Style* (Podstawy stylu programowania). Wydałem ją w 1974 roku wraz z P.J. „Billem" Plaugerem, który zajmował jeden z sąsiednich gabinetów. Postanowiliśmy z Billem pójść w ślady klasycznego podręcznika Strunka i White'a *The Elements of Style* (Podstawy stylu)[4] i przedstawiać fragmenty źle napisanego kodu wraz z wyjaśnieniami, jak można ulepszyć każdy z nich.

Nasz pierwszy przykład zaczerpnęliśmy z książki, którą pokazał mi Dick. Któregoś dnia wszedł do mojego gabinetu z publikacją na temat analizy numerycznej, zły jak osa z powodu niskiej jakości fragmentów poświęconych danym cyfrowym. Zerknąłem na okropny fragment kodu w Fortranie:

[4] Podręcznik stylistycznego pisania w języku angielskim, którego pierwsze wydanie ukazało się w 1920 roku — *przyp. tłum.*

```
      DO 14 I=1,N
      DO 14 J=1,N
14 V(I,J)=(I/J)*(J/I)
```

Jeśli nie programowałeś w Fortranie, pozwól mi na kilka słów wyjaśnienia. Powyższy kod składa się z dwóch zagnieżdżonych pętli `DO`, kończących się na linii opatrzonej etykietą `14`. W ramach każdej pętli zmienna przybiera rosnące wartości od dolnej granicy przedziału do granicy górnej, co oznacza, że w pętli zewnętrznej zmienna `I` przyjmuje wartości od `1` do `N` i dla każdej z tych wartości `I`, w pętli wewnętrznej zmienna `J` przyjmuje wartości od `1` do `N`. Zmienna `V` jest tablicą o `N` wierszach i `N` kolumnach; pętla `I` przegląda poszczególne wiersze, a dla każdego wiersza pętla `J` przegląda poszczególne kolumny.

Ta konkretna para pętli tworzy zatem macierz o wymiarach `N` na `N`, w której po przekątnej znajdują się wartości 1, a we wszystkich pozostałych polach wartości 0, jak w tym przypadku, gdy `N` wynosi 5:

```
1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1
```

Działanie kodu opiera się na fakcie, że operacja dzielenia liczb całkowitych w Fortranie odrzuca część ułamkową, jeśli więc `I` nie jest równe `J`, jedno z działań dzielenia zwróci wartość 0, ale jeśli `I` jest równe `J` (jak w przypadku przekątnej macierzy), wynik będzie wynosił `1`.

Wszystko to wydało mi się aż zbyt sprytne, a źle rozumiany spryt jest w programowaniu bardzo złym pomysłem.

Przepisanie tego kodu w prostszy i oczywisty sposób pozwoliło uzyskać znacznie czytelniejszą wersję: dla każdego przebiegu zewnętrznej pętli, wewnętrzna pętla nadaje wszystkim elementom wiersza `I` wartość 0, a potem zewnętrzna pętla nadaje ukośnym elementom `V(I,I)` wartość 1:

```
C STWÓRZ MACIERZ JEDNOSTKOWĄ V
      DO 14 I = 1,N
         DO 12 J = 1,N
  12 V(I,J) = 0.0
  14 V(I,I) = 1.0
```

Stało się to podstawą naszej pierwszej reguły stylu programowania:

*Pisz przejrzyście — nie bądź za sprytny.*

W 1976 roku Dick odszedł z Bell Labs i przeszedł do Naval Postgraduate School w Monterey w stanie Kalifornia, gdzie uczył do ostatnich dni życia — zmarł na początku 1998 roku w wieku 82 lat. Ponoć jeden z jego kursów studenci nazwali „Hamming on Hamming"[5], co stanowi specyficzną paralelę dla tej części książki.

Dick bez przerwy zastanawiał się nad wszystkim co robi — i dlaczego. Często mawiał, że „celem informatyki są wnioski, nie liczby", i nawet miał krawat z takim napisem (po chińsku). Między innymi bardzo wcześnie doszedł do wniosku, że komputery wkrótce będą wykonywać połowę prac w Bell Labs. Wtedy żaden z kolegów się z nim nie zgodził, lecz po niedługim czasie okazało się, że jego szacunki były nawet zaniżone. Mawiał, że piątkowe popołudnia są wymarzoną porą na świetne pomysły, siedział więc i myślał, choć zawsze chętnie przyjmował gości, takich jak ja.

Kilka lat po odejściu na emeryturę Dick wygłosił interesującą prelekcję, która zawierała kwintesencję jego rad dotyczących udanej kariery zawodowej. Mowę tę, zatytułowaną „You and Your Research" (Ty i twoje badania) można znaleźć w internecie. Pierwszą wersję prelekcji przedstawił w marcu 1986 roku w Bellcore. Wysłuchałem jej dzięki Kenowi Thompsonowi, który podrzucił mnie na miejsce. Od kilkudziesięciu lat polecam studentom zapoznanie się z tym materiałem. Naprawdę warto przeczytać transkrypcję albo obejrzeć jedną z dostępnych wersji filmowych.

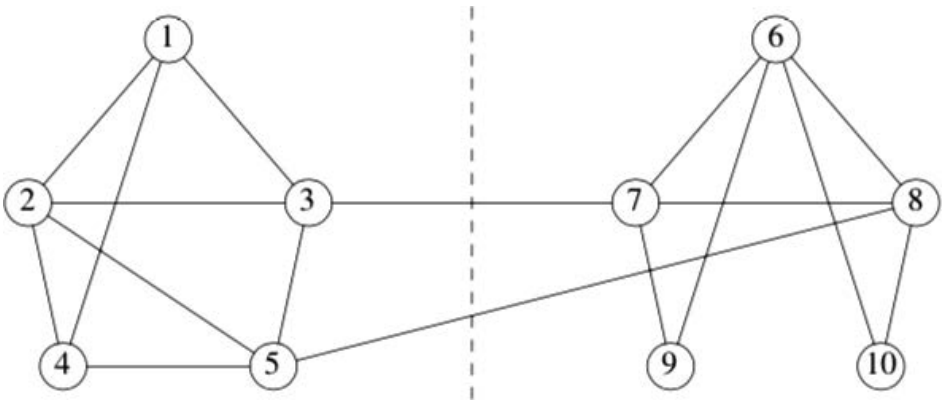
Latem 1967 roku gabinet po drugiej stronie korytarza zajmował Vic Vyssotsky (rysunek 1.6), kolejny niezwykle inteligentny i utalentowany programista. Wspólnie z Corbym Vic odpowiadał za tę część systemu Multics, którą zajmowało się Bell Labs, lecz pomimo nawału pracy niemal codziennie udawało mu się znaleźć chwilę na rozmowę ze skromnym stażystą. Vic namówił mnie na prowadzenie kursu Fortrana dla fizyków i chemików, którzy potrzebowali nauczyć się programowania. Uczenie języka osób, które nigdy nie miały do czynienia z kodem sprawiło mi zaskakującą frajdę. Dzięki tym zajęciom przełamałem tremę przed wystąpieniami publicznymi i przetarłem sobie szlak do wielu późniejszych nauczycielskich występów.

[5] Specyficzna gra słów związana z nazwiskiem naukowca; *hamming* oznacza zbytnią emfazę i wczuwanie się w rolę — *przyp. tłum.*

**Rysunek 1.6.** *Vic Vyssotsky, ok. 1982. (Dzięki uprzejmości Bell Labs)*

Wkrótce potem Vic przeniósł się do innego działu Bell Labs, gdzie pracował nad systemem obrony przeciwrakietowej Safeguard. Ostatecznie wrócił jednak do Murray Hill i został dyrektorem wykonawczym odpowiedzialnym za badania informatyczne, a tym samym moim przełożonym o kilka szczebli wyżej.

Wiosną 1968 roku zacząłem pracować nad zagadnieniem, któremu miała być poświęcona moja praca doktorska. Temat zasugerował mój promotor, Peter Weiner — chodziło o *partycjonowanie grafu*: mając zbiór punktów węzłowych połączonych liniami, należało znaleźć sposób na podzielenie węzłów na dwie równe grupy tak, by liczba linii łączących węzeł należący do jednej grupy z węzłem z drugiej grupy była jak najmniejsza. Rysunek 1.7 przedstawia przykład rozwiązania tego problemu: dowolny inny podział węzłów na grupy po pięć wymagałby więcej niż dwóch linii między zbiorami.

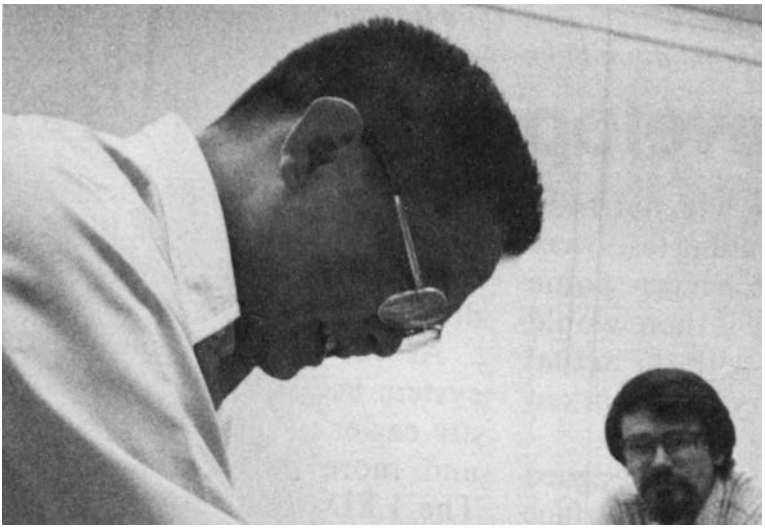



**Rysunek 1.7.** *Przykład partycjonowania grafu*

Problem ten miał rzekomo praktyczne podstawy: jak przypisać części programu do stron pamięci w taki sposób, aby po uruchomieniu programu ograniczyć przesyłanie stron do pamięci i z powrotem. Węzły reprezentowały bloki kodu, a krawędzie — możliwe przejścia między blokami. Dla każdej krawędzi można było określić wagę, która odzwierciedlała częstotliwość przejść, a tym samym koszt związany z umieszczeniem dwóch bloków programu na różnych stronach.

Zagadnienie było poniekąd sztuczne, lecz stanowiło wiarygodne odzwierciedlenie czegoś realnego; istniały też inne, konkretne problemy, które opierały się na tym samym abstrakcyjnym modelu. Na przykład, jak należy rozmieścić komponenty na płytkach drukowanych, aby do minimum ograniczyć liczbę i długość ścieżek łączących te płytki? Albo — choć ta kwestia nie jest już może tak przekonująca — jak rozlokować pracowników na piętrach budynku, aby na tych samych piętrach znaleźli się ci, którzy najczęściej ze sobą rozmawiają?

Zagadnienie nadawało się na temat pracy doktorskiej, ale nie robiłem postępów. Po powrocie do Bell Labs na drugi staż, latem 1968 roku, przedstawiłem problem Shenowi Linowi (rysunek 1.8), który niedługo wcześniej opracował najskuteczniejszy ze znanych algorytmów rozwiązujących klasyczny problem komiwojażera: przy danej liczbie miast, należy znaleźć najkrótszą trasę pozwalającą odwiedzić każde z nich dokładnie raz i wrócić do domu.



**Rysunek 1.8.** *Shen Lin, ok. 1970 roku. (Dzięki uprzejmości Bell Labs)*

Shen wpadł na obiecujący pomysł rozwiązania problemu z partycjonowaniem grafu, choć nie gwarantował, że pozwoli on osiągać najlepsze rezultaty, a mnie udało się go efektywnie zaimplementować. Przeprowadziłem eksperymenty na dużej liczbie grafów, aby się przekonać, na ile skutecznie sprawdza się ów algorytm w praktyce. Wydawał się bardzo dobry, ale nigdy nie odkryliśmy sposobu gwarantującego uzyskanie optymalnego rozwiązania. Przy okazji znalazłem kilka ciekawych przypadków szczególnych grafów, umożliwiających opracowanie innych algorytmów, które były szybkie i takie rozwiązania gwarantowały. Łącznie, wszystkie osiągnięte rezultaty wystarczały do napisania rozprawy i pod koniec lata miałem już wszystko, czego mi było trzeba. Napisałem ją jesienią, a końcowy egzamin ustny przeszedłem w styczniu 1969 roku. (Optymistyczne szacunki uczelni w Princeton o ukończeniu doktoratu w trzy lata zostały przekroczone o półtora roku).

Tydzień później podjąłem pracę w Computing Science Research Center w Bell Labs. Obyło się bez rozmowy kwalifikacyjnej; któregoś jesiennego dnia firma po prostu wysłała mi propozycję zatrudnienia, z jednym zastrzeżeniem: ukończenia rozprawy doktorskiej. Sam Morgan, kierownik Centrum i zarazem mój przełożony o dwa szczeble wyżej w hierarchii, powiedział mi: „Nie zatrudniamy niedorobionych doktorantów". Ukończenie pracy zdecydowanie się opłacało: w grudniu dostałem kolejny list, tym razem z informacją o niemałej podwyżce — i to zanim jeszcze zgłosiłem się do pracy!

Nawiasem mówiąc, choć Shen i ja nie zdawaliśmy sobie wtedy z tego sprawy, istniał powód, który uniemożliwiał nam znalezienie efektywnego algorytmu dzielenia grafu, dającego najlepsze możliwe rozwiązanie w każdym przypadku. Inni także łamali sobie głowy nad trudnościami związanymi z optymalizacją zagadnień kombinatorycznych, takich jak podział grafu, i odkryli kilka ciekawych zależności o charakterze ogólnym.

W ramach jednego z takich niezwykłych spostrzeżeń, w 1971 roku Stephen Cook, matematyk i informatyk z Uniwersytetu w Toronto wykazał, że wiele tego rodzaju trudnych zagadnień — w tym podział grafu — jest tożsamych w tym znaczeniu, że gdyby udało się nam znaleźć efektywny algorytm (czyli coś lepszego niż sprawdzanie wszystkich możliwych rozwiązań) dla jednego z nich, znaleźlibyśmy efektywne algorytmy dla wszystkich. Kwestia skali trudności tych problemów pozostaje w informatyce otwarta, zakłada się jednak, że istotnie są trudne. W 1982 roku Cook otrzymał za swoje prace Nagrodę Turinga.

Gdy w 1969 roku trafiłem do Bell Labs już jako etatowy pracownik, nikt nie powiedział mi, czym się mam zajmować. To było typowe podejście: człowiek

poznawał innych, był zachęcany do rozglądania się i zdany na siebie, jeśli chodzi o znalezienie własnych tematów badań i współpracowników. Z dzisiejszej perspektywy wydaje się to onieśmielająco trudne, lecz nie przypominam sobie, abym się tym przejmował. Działo się tak wiele, że nietrudno było znaleźć temat do zgłębienia czy kogoś do współpracy, i po dwóch latach znałem już ludzi oraz część realizowanych w centrum przedsięwzięć.

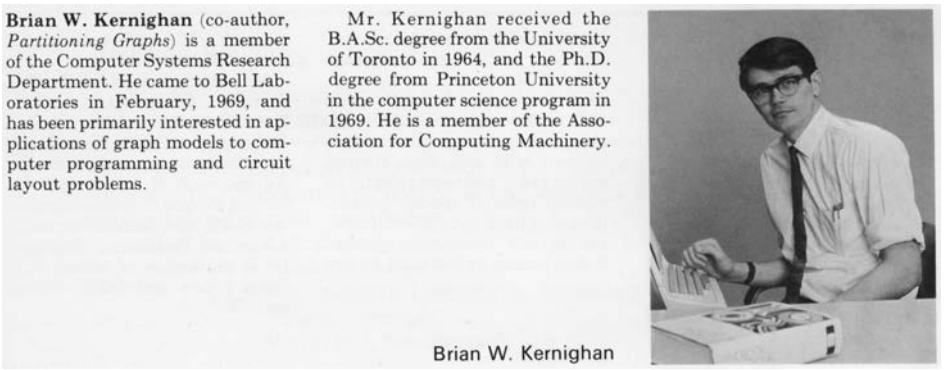
Ten brak wyraźnych wskazówek ze strony przełożonych był standardową praktyką. Projekty w centrum 1127 nie były przydzielane przez kierownictwo, ale wyrastały oddolnie i gromadziły wokół siebie ludzi zainteresowanych danym tematem. To samo dotyczyło pracy w innych działach Bell Labs: jeśli brałem udział w pracach jakiegoś zespołu, mogłem zachęcić kolegów badaczy do przyłączenia się (aczkolwiek byli wtedy traktowani jako wolontariusze).

Tak czy inaczej, przez pewien czas wspólnie z Shenem pracowałem nad problemami z dziedziny kombinatoryki. Shen miał wyjątkową głowę do tych zagadnień i potrafił intuicyjnie wskazać obiecujące podejścia dzięki samodzielnemu przeanalizowaniu niewielkiej puli przykładów. Wpadł na nowy pomysł rozwiązania problemu komiwojażera, pozwalający udoskonalić poprzedni algorytm (który i tak był już najlepszym z wówczas znanych), a ja zaimplementowałem go w Fortranie. Rozwiązanie działało i przez wiele lat było uznawane za niedoścignione.

Tego rodzaju praca sprawiała przyjemność i dawała satysfakcję, ale choć całkiem nieźle potrafiłem przekładać koncepcje na działający kod, w kwestiach wymyślania algorytmów nie byłem zbyt dobry. Stopniowo zacząłem więc realizować się w innych obszarach: tworzeniu programów do przygotowywania dokumentów, opracowywaniu specjalistycznych języków programowania oraz — w niewielkim zakresie — w pisaniu.

Potem jeszcze kilkakrotnie współpracowałem z Shenem, między innymi podczas tworzenia skomplikowanego narzędzia służącego do optymalizowania prywatnych sieci klientów AT&T. Dobrze było przechodzić w tę i z powrotem między stosunkowo czystą informatyką a systemami, które miały dla firmy konkretną, praktyczną wartość.

Dział *public relations* Bell Labs był zachwycony pracami Shena nad rozwiązaniem problemu komiwojażera, dzięki czemu Shen trafił do wielu materiałów reklamowych firmy. Nieostra fotka z rysunku 1.8 została zaczerpnięta z jednego z nich (stoję w rogu), a rysunek 1.9 pochodzi z jakiegoś wydanego na kredowym papierze magazynu promocyjnego, wydawanego przez Bell Labs — była w nim mowa o naszej pracy nad podziałem grafów; zapewne miało to miejsce po uzyskaniu patentu na algorytm.



**Brian W. Kernighan** (co-author, *Partitioning Graphs*) is a member of the Computer Systems Research Department. He came to Bell Laboratories in February, 1969, and has been primarily interested in applications of graph models to computer programming and circuit layout problems.

Mr. Kernighan received the B.A.Sc. degree from the University of Toronto in 1964, and the Ph.D. degree from Princeton University in the computer science program in 1969. He is a member of the Association for Computing Machinery.

Brian W. Kernighan

**Rysunek 1.9.** *Zdjęcie z działu PR, ok. 1970. (Dzięki uprzejmości Bell Labs)*

Pragnę zauważyć, że na wzmiankowanym zdjęciu jestem pod krawatem, co było u mnie rzeczą niebywałą. Kilka lat później razem z Dennisem Ritchiem napisaliśmy o języku C artykuł dla innego magazynu firmowego, o ile pamiętam był to „Western Electric Engineer". Przed publikacją poproszono nas o wysłanie zdjęć, które miały zostać zamieszczone w artykule, co też uczyniliśmy. Po kilku tygodniach powiedziano nam, że zdjęcia zostały zgubione. Odparliśmy, że to żaden kłopot, bo możemy wysłać je jeszcze raz, na co usłyszeliśmy: „Czy tym razem moglibyście założyć krawaty?". Wkrótce po stanowczej odmowie magazyn się ukazał, a w nim nasze oryginalne, „nieukrawacone" fotografie, które czarodziejsko się odnalazły.

Po rozpoczęciu etatowej pracy otrzymałem biuro na czwartym piętrze budynku 2, na korytarzu nieopodal klatki schodowej numer 9. Stało się ono moją siedzibą na 30 lat — niezmiennym punktem w świecie nieustannych zmian. Przez te lata wśród moich sąsiadów byli Ken Thompson, Dennis Ritchie, Bob Morris, Joe Ossanna i Gerard Holzmann oraz wybitni goście, jak John Lions, Andy Tanenbaum i David Wheeler.

Przez ostatnią dekadę mojej pracy w Bell Labs, biura Kena Thompsona i Dennisa Ritchiego mieściły się dokładnie naprzeciwko mojego, po drugiej stronie korytarza. Rysunek 1.10 przedstawia biuro Dennisa, sfotografowane w październiku 2005 roku z progu mojego starego gabinetu. Pokój Kena znajdował się po lewej stronie.

Na przestrzeni lat bezpośrednio sąsiadowałem między innymi z Billem Plaugerem, Lorindą Cherry, Peterem Weinbergerem i Alem Aho, a zaledwie kilka kroków dzieliło mnie od pracowni Douga McIlroya, Roba Pike'a oraz Rona Bentleya. Łatwo jest współpracować z osobami, które znajdują się fizycznie blisko, pod względem sąsiedztwa miałem więc wiele szczęścia.

**Rysunek 1.10.** *Biuro Dennisa Ritchiego w 2005 roku*

## 1.5. 137 → 127 → 1127 → 11276

Kim byli w owym czasie główni gracze i jak wyglądało otoczenie? Na początku lat 70. w ośrodku Computing Science Research Center pracowało tylko nieco ponad trzydzieści osób, z czego bodaj cztery lub najwyżej sześć zajmowało się Uniksem lub blisko związanymi z nim kwestiami. Rysunek 1.11 przedstawia wycinki z wewnętrznej książki telefonicznej Bell Labs. Kartki nie pożółkły ze starości; gdy trafiłem tam do pracy, telefony firmowe były drukowane na żółtym papierze, charakterystycznym dla starych książek telefonicznych.

Kartki pochodzą z 1969 roku. Kierownikiem Computing Science Research Center był wówczas Sam Morgan (rysunek 1.12), wybitny znawca matematyki stosowanej i ekspert w dziedzinie teorii komunikacji. Doug McIlroy, który odegrał w rozwoju Uniksa niezwykle ważną rolę — choć mało kto zdawał sobie z niej sprawę — kierował zespołem, do którego należał Ken Thompson i kilka innych osób biorących udział we wczesnych pracach nad Uniksem; między innymi Rudd Canaday, Bob Morris, Peter Neumann i Joe Ossanna. W dziale Elliota Pinsona pracowali Dennis Ritchie, Sandy Fraser i Steve Johnson, którzy także przez wiele lat uczestniczyli w rozwijaniu Uniksa.

**Computing Science Research Center**

| | | |
|---|---|---|
| 137 | Morgan S P, Director, Computing Science Research Center | MH 6490 |
| | Kalainikas Miss E, Secretary | MH 6491 |
| 1371 | McIlroy M D, Head, Computing Techniques Research Department | MH 6050 |
| | Marky Miss G A, Secretary | MH 6051 |
| | Dimino L A | MH 2390 |
| | Aho A V | MH 4862 |
| | Canaday R H | MH 3038 |
| | Friedman A D | MH 4716 |
| | Jensen P D | MH 6292 |
| | Knowlton K C | MH 2328 |
| | Menon P R | MH 2736 |
| | Morris R | MH 3878 |
| | Neumann P G | MH 2666 |
| | Ossanna J F | MH 3520 |
| | Thompson K L | MH 2394 |
| | Ullman J D | MH 6627 |
| | Wagner Mrs M R | MH 2879 |
| | Weiss Miss R A | MH 2007 |
| 1373 | Pinson E N, Head, Computer Systems Research Department | MH 2582 |
| | Blejwas Miss V M, Secretary | MH 2583 |
| | Fraser A G | MH 3685 |
| | Johnson S C | MH 3968 |
| | Kernighan B W | MH 6021 |
| | Ritchie D M | MH 3770 |
| | Sturman J N | MH 3164 |
| | Winikoff A W | MH 2661 |
| 1374 | Brown W S, Head, Computing Mathematics Research Department | MH 4822 |
| | Blejwas Miss V M, Secretary | MH 4823 |
| | Hall A D | MH 4006 |
| | Goldstein A J, Supervisor, Mathematical Techniques Group | MH 2655 |
| | Lin S | MH 2111 |
| | Shafer D M | MH 6862 |
| 1374 | Traub J F, Supervisor, Numerical Mathematics Group | MH 2383 |
| | Businger P A | MH 2059 |
| | Richman P L | MH 3932 |
| | Schryer N L | MH 2912 |
| 1376 | Hamming R W, Head, Computing Science Research Department | MH 2064 |
| | Marky Miss G A, Secretary | MH 2065 |

**Rysunek 1.11.** *Książka telefoniczna Bell Labs, ok. 1969 roku. (Dzięki uprzejmości Gerarda Holzmanna)*

**Rysunek 1.12.** *Sam Morgan, kierownik centrum 1127, ok. 1981 roku. (Dzięki uprzejmości Gerarda Holzmanna)*

Choć większość badaczy miała stopień doktora, nikt nie zwracał się do siebie per „doktorze", tylko po imieniu. Jedynymi wyjątkami jeśli chodzi o tytuły widoczne w książce telefonicznej z rysunku 1.11 były panie, u których odnotowano kwestię stanu cywilnego (*Mrs* lub *Miss*); u mężczyzn takich informacji nie podawano. Nie pamiętam, kiedy dokładnie przestano stosować te adnotacje, lecz na początku lat 80. na pewno już ich nie było.

W latach 60. i 70. na stanowiskach technicznych w Bell Labs niewiele było kobiet i osób kolorowych; większość kadry inżynierskiej stanowili biali mężczyźni i stan ten utrzymywał się jeszcze długo. Pod tym względem firma Bell Labs nie różniła się od większości środowisk technicznych w tamtej epoce historii informatyki.

Na początku lat 70. w Bell Labs uruchomiono trzy długofalowe programy, które miały na celu poprawę tej sytuacji. W 1972 roku ruszył Cooperative Research Fellowship Program (CRFP), w ramach którego co roku około 10 studentów należących do mniejszości etnicznych mogło uzyskać środki na cztery lata (lub więcej) studiów podyplomowych, potrzebnych do uzyskania tytułu doktora. Z kolei Graduate Research Program for Women (GRPW), który uruchomiono w 1974 roku, zapewniał analogiczne wsparcie kobietom — mogło na nie liczyć mniej więcej 15 – 20 pań rocznie. Kilka spośród nich w różnych okresach pracowało w centrum 1127 i na moim wydziale, a większość z powodzeniem kontynuowała kariery w Bell Labs, na uniwersytetach i w innych firmach. Ponadto każdego roku, w ramach Summer Research Program (SRP), który również wystartował w 1974 roku, mniej więcej sześćdziesięciu studiującym kobietom i przedstawicielom mniejszości etnicznych zapewniano w pełni opłacone staże letnie. Prowadzono je w Murray Hill, Holmdel i niekiedy innych miejscach, na zasadzie kontaktu z indywidualnie przypisanym do danej osoby badaczem-mentorem. Uczestniczyłem w programie SRP w centrum 1127 przez ponad piętnaście lat i przez ten czas poznałem wielu błyskotliwych studentów, a dla kilku pełniłem funkcję mentora.

W długiej perspektywie programy te okazały się skuteczne, lecz na przestrzeni lat 60. i 70. środowisko wciąż było dość monolityczne i jestem przekonany, że nie w pełni zdawałem sobie sprawę z niektórych konsekwencji tego faktu.

W Bell Labs panowała prosta hierarchia kierownicza. Najwyższe stanowisko piastował prezes, który miał pod sobą 15 – 25 tysięcy osób. Szczebel niżej plasowały się poszczególne główne piony działalności, oznaczone numerami 10 (badania), 20 (rozwój), 50 (centrale telefoniczne), 60 (systemy wojskowe) i tak dalej,

a każdy z nich miał swojego szefa w randze wiceprezesa. Pion badań z kolei dzielił się na fizykę (11), matematykę i systemy komunikacyjne (13), chemię (15) i podobne — każda z tych jednostek miała dyrektora wykonawczego oraz doradców patentowych i prawnych. Ośrodek badań matematycznych (Mathematics Research) miał numer 131, a ośrodek badań obliczeniowych (Computing Science Research), nazywany „centrum 137", był podzielony na kilka działów w rodzaju 1371. Kilka lat później, w ramach poważnej reorganizacji, wszystkie te oznaczenia zostały zmienione i w rezultacie staliśmy się „centrum 127"; potem zaś, wskutek kolejnych zmian, numer ten poprzedzono jeszcze jedną cyfrą — i tak otrzymaliśmy oznaczenie 1127, które funkcjonowało do 2005 roku, czyli jeszcze całkiem długo po moim przejściu na emeryturę w roku 2000.

Hierarchia nie była zbyt rozbudowana. Badacze tacy jak ja byli nazywani „członkami kadry technicznej" lub w skrócie MTS (od ang. *member of technical staff*). MTS na ogół dostawali prywatne pracownie, choć powszechnie oczekiwało się, by przez większość czasu drzwi były otwarte. Byliśmy nadzorowani, choć w ciągu tych wszystkich lat centrum 1127 nie miało wielu nadzorców. Kolejny szczebel stanowił szef działu — osoba w rodzaju Douga McIlroya, odpowiedzialna za kilku lub kilkunastu badaczy. Na następnym szczeblu drabiny znajdował się kierownik centrum, który przewodził kilku działom, na kolejnym dyrektor wykonawczy panujący nad kilkoma centrami, a przełożonym dyrektorów wykonawczych był wiceprezes.

Wiceprezesi bezpośrednio podlegali prezesowi. Bill Baker, znakomity chemik, był wiceprezesem pionu badań w latach 1955 – 1973, a potem prezesem Bell Labs aż do 1980 roku. Gdy piastował stanowisko wiceprezesa mawiało się, że zna wszystkich MTS z nazwiska i wie, czym się zajmują. Sądzę, że mogła to być prawda, zawsze wiedział bowiem, nad czym pracują moi koledzy i ja.

Szeregowym członkiem kadry technicznej byłem do roku 1981, gdy ostatecznie ugiąłem się pod presją i przyjąłem nominację na szefa działu. Większość ludzi niechętnie obejmowała stanowiska kierownicze, bo choć nie oznaczało to końca własnych badań, wiązało się z ich znacznym spowolnieniem oraz z odpowiedzialnością wynikającą z konieczności doglądania prac działu, co bywało trudne. Propozycjom towarzyszyły oczywiście typowe argumenty: „To nieuniknione, więc czemu nie teraz?". Słyszało się też coś wręcz przeciwnego: „To może być twoja ostatnia szansa". Albo: „Jeśli nie ty, przejmie to ktoś inny, nie tak dobry".

Tak czy inaczej, zostałem zatem szefem nowego działu o numerze 11276 i bezpiecznie nieokreślonej nazwie „Computing Structures Research" (Badanie struktur

obliczeniowych). W dziale tym pracowało zwykle 8 – 10 osób o przytłaczająco obszernym spektrum zainteresowań: sprzęt graficzny, narzędzia do projektowania układów scalonych, przygotowywanie dokumentacji, systemy operacyjne, sieci, kompilatory, C++, projektowanie systemów bezprzewodowych, geometria obliczeniowa, teoria grafów, złożoność obliczeniowa i wiele wątków pobocznych. Zapoznanie się z pracami każdej z tych osób na tyle dobrze, by móc je następnie przybliżyć przełożonym, zawsze było wyzwaniem, ale zarazem dawało satysfakcję, a wiele z tego, czego się wtedy nauczyłem, zapamiętałem na zawsze.

Kolejne szczeble w hierarchii kierowniczej wiązały się z określonymi uposażeniami. Niektóre były oczywiste, takie jak coraz większe gabinety przysługujące na poziomie szefa działu i wyżej. O ile kojarzę, szefowie działów dostawali też podwyżkę, ale najwyraźniej nie była bardzo wysoka, bo nie zapadła mi w pamięć.

Inne korzyści były subtelniejsze: począwszy od szefów działów, gabinety miały dywany, podczas gdy zwykli pracownicy musieli zadowolić się gołym linoleum albo winylowymi płytkami. Po awansie otrzymałem wydrukowaną na błyszczącym papierze broszurę, z której mogłem wybrać kolor dywanu, meble biurowe i tak dalej. Przez krótki czas eksperymentowałem z nowym biurkiem, ale okazało się za duże i niewygodne, wróciłem więc do zabytkowego biurka marki Steelcase, które odziedziczyłem w 1969 roku. Z dywanu zaś zrezygnowałem w ogóle, gdyż nieszczególnie były mi w smak wyróżnienia związane z rangą. Sam Morgan usilnie przekonywał mnie do dywanu. Twierdził, że któregoś dnia zacznie mi zależeć na autorytecie wiążącym się z jego posiadaniem. Mimo wszystko twardo odmawiałem i kwestia dywanowych dystynkcji ostatecznie zeszła na dalszy plan.

Zasadniczym, corocznym zadaniem szefów działów była analiza pracy podwładnych w ramach wyrafinowanego rytuału zwanego „przeglądem wartości". Raz do roku każdy MTS spisywał na jednej stronie jednej kartki podsumowanie swoich dokonań w ciągu tego roku; w centrum 1127 kartki te funkcjonowały pod nazwą *I am great report*[6], którą — jak sądzę — wymyślił Sam Morgan. Następnie szef działu sporządzał dokument, który stanowił podsumowanie i ocenę pracy podopiecznych, z uwzględnieniem tak zwanych obszarów do usprawnienia — ta sekcja dokumentu z założenia miała zawierać konstruktywną krytykę.

Pisanie tych analiz i przekazywanie opinii było trudne, istniała więc silna tendencja do pomijania sekcji z obszarami do usprawnienia. Któregoś roku powiedziano

[6] Nazwę tę można rozumieć dwojako: „jestem świetnym raportem" i „raport o tym, jaki jestem świetny" — *przyp. tłum.*

nam jednak, że trzeba ją wypełnić; uniki w rodzaju zostawiania wolnego miejsca albo adnotacji „nie dotyczy" przestały wchodzić w rachubę. Wymyśliłem, że będę wpisywać: „Świetna robota, tak trzymać"; komentarz ten jakimś cudem rok albo dwa przechodził bez echa — dopiero potem usłyszałem, że potrzebne są bardziej krytyczne uwagi, ponieważ nikt nie jest doskonały. Na szczęście nie musiałem tego robić w odniesieniu do gwiazd pokroju Kena Thompsona. Co niby miałbym napisać?

Szefowie działów spotykali się z kierownikiem centrum, aby dojść do porozumienia w sprawie oceny każdego MTS. Zwykle wymagało to całodziennych dyskusji. Kilka tygodni później organizowano następne spotkanie, również całodniowe, w trakcie którego określano wysokość wypłat na kolejny rok poprzez przydzielenie każdemu MTS części wspólnej puli na podwyżki. Te dwie związane ze sobą oceny miały swoje oficjalne nazwy: „przegląd wartości" i „przegląd płac", ale ja zawsze nazywałem je w myślach przeglądami abstrakcji i konkretów.

Ten proces był powtarzany na wyższych szczeblach hierarchii — dyrektor wykonawczy przeglądał wyniki wszystkich MTS wspólnie z kierownikami centrów, a przy okazji oceniał dokonania szefów działów.

W niektórych centrach przegląd wartości stawał się polem do rywalizacji, lecz nasze przeglądy miały wybitnie koleżeński charakter. Nie kierowano się podejściem „moi ludzie są lepsi niż twoi", lecz raczej „nie zapominajmy o innej dobrej robocie, którą wykonał twój człowiek".

Być może patrzę przez różowe okulary, lecz moim zdaniem całość zdawała egzamin, bo przełożeni aż do najwyższych szczebli mieli kompetencje techniczne, a każdy oswajał się z tym procesem, będąc na niższych poziomach w hierarchii. System raczej nie faworyzował ani praktyki, ani teorii — a przynajmniej tak było w centrum 1127. W równym stopniu ceniło się dobre programy i dobre publikacje. Brak odgórnych propozycji albo planów na przyszłe przedsięwzięcia stanowił zaletę. Oczekiwano wprawdzie, że pod koniec roku każdy będzie mógł się poszczycić osiągnięciami na miarę dwunastu miesięcy pracy, lecz większość falstartów ignorowano, a kadra zarządzająca brała długoterminową poprawkę na osoby, które przez kilka lat zajmowały się jednym tematem. Moim zdaniem pomagało także to, że w pionie badań niewiele było poziomów kierowniczych, ludzie zasadniczo nie zastanawiali się więc nad awansami. Dla kogoś, komu zależało na stanowisku menedżerskim, lepszym wyjściem było znalezienie sobie miejsca poza strukturami działu badań.

Ciekawe wnioski płyną z porównania procesu oceniania pracownika w Bell Labs do analogicznych mechanizmów stosowanych na uczelniach naukowych. W przypadku tych drugich, ogromny wpływ na zatrudnienie — a szczególnie

na awans — ma zgromadzenie co najmniej kilkunastu listów polecających od cenionych naukowców spoza placówki, którzy zajmują się tą samą dziedziną wiedzy. Takie podejście zachęca do pogłębiania wiedzy w bardzo wąskim obszarze, ponieważ celem ocenianego jest opanowanie jakiejś specjalizacji w takim stopniu, by recenzenci mogli uczciwie stwierdzić: „Ten człowiek, na tym etapie swojej kariery, jest w tej wąskiej specjalizacji najlepszy".

Dla odmiany w Bell Labs tworzono swego rodzaju oddolny ranking obejmujący wszystkich badaczy. Szef każdego działu oceniał swoich ludzi; oceny te były następnie scalane przez wszystkich szefów działów w danym centrum, a rezultaty poddawano podobnej operacji na dwóch następnych szczeblach, dzięki czemu każdy mógł w przybliżeniu ocenić swoją pozycję w ramach całej organizacji.

Osoba o wybitnych osiągnięciach w wąskiej dziedzinie mogła być wysoko oceniana przez swoich bezpośrednich przełożonych, lecz było niezmiernie mało prawdopodobne, że wyniki tej pracy będą znane na kolejnych szczeblach. Dla odmiany przedsięwzięcia interdyscyplinarne były bardziej widoczne „na górze", bo słyszało o nich więcej menedżerów. Im szerzej zakrojona była współpraca, tym głośniejszym echem odbijała się wśród kierownictwa. W efekcie organizacja jako całość bardzo sprzyjała współpracy i badaniom interdyscyplinarnym. A ponieważ decydenci na poszczególnych szczeblach przeszli przez ten sam proces, mieli podobne inklinacje.

Kierowałem działem przez ponad 15 lat i — jak sądzę — byłem w najlepszym razie przeciętnym szefem, ucieszyłem się więc na wieść o możliwości ustąpienia ze stanowiska. Inni z powodzeniem unikali awansu przez wiele lat; Dennis Ritchie został szefem działu na długo po mnie, a Kena Thompsona zaszczyt ten ominął zupełnie.

Po dwudziestu latach wykładania na uczelni wciąż nie jestem zwolennikiem oceniania cudzej pracy. Niestety, jest to jednak konieczne i czasami należy podjąć niewygodną decyzję, która wpłynie na czyjeś życie — trzeba na przykład kogoś zwolnić (czego na szczęście nigdy nie musiałem robić) lub oblać studenta (co nie jest częste, ale się zdarza). Jedną z zalet procesu w Bell Labs był fakt, że opierał się on na wspólnej ocenie dokonywanej przez ludzi, którzy rozumieli charakter wykonywanej pracy. Jak powiedział Doug McIlroy, „Geniusz tego systemu tkwił w koleżeństwie. Niczyj awans nie był uzależniony od relacji z tylko jednym szefem". Stosowany w Bell Labs system oceniania nie był doskonały, ale sprawdzał się zupełnie dobrze — słyszałem i czytałem o innych, znacznie gorszych systemach okresowej oceny pracowników.

**UNIX NARODZIŁ SIĘ W 1969 ROKU** na ostatnim piętrze siedziby Bell Labs. Stworzyło go dwóch pasjonatów z pomocą niewielkiej grupy współpracowników i sympatyków. Od chwili powstania ten system operacyjny cieszył się ogromną popularnością. Doprowadził do wielu przełomów i na zawsze zmienił bieg historii informatyki. Dziś Unix i jego pochodne stanowią serce wielu systemów i usług. Dzięki nim działają Google, Facebook, Amazon, produkty Apple oraz smartfony z Androidem, a także asystent głosowy Alexa. Codziennie Twoje zachowanie w internecie jest uważnie śledzone, spotykasz się z personalizacją reklam — to również dzieje się dzięki systemom uniksowym.

**TA KSIĄŻKA** jest unikalną relacją historyczną z początków Uniksa. Zawiera wspomnienia naocznych świadków z pierwszych lat istnienia tego systemu, jego rozwoju i stopniowego przenikania technologii uniksowych do poszczególnych dziedzin informatyki. Znajdziesz tu niewiele trudnych, naukowych i technicznych detali, za to zachwycisz się niezwykłą historią, zapoczątkowaną przez wyjątkowych ludzi, którzy wybiegłszy myślą daleko w przyszłość, zapewnili środowisko do nieskrępowanych, niekonwencjonalnych eksperymentów. Przekonasz się, że dzieje Uniksa wciąż są skarbnicą inspiracji w zakresie projektowania oprogramowania i nowych możliwości technologii.

## W KSIĄŻCE:

- **co się zdarzyło w Bell Labs w połowie XX wieku**
- **jak powstał proto-Unix**
- **pierwsze wersje Uniksa**
- **ewolucja i powstawanie powiązanych technologii**
- **Minix, Linux i inni potomkowie**
- **czym jest rzeczywiste dziedzictwo Uniksa**

## CZY NIEWIARYGODNY SUKCES UNIKSA BYŁ TYLKO DZIEŁEM SPLOTU OKOLICZNOŚCI?

**Brian W. Kernighan** wykłada na Uniwersytecie Princeton. Wcześniej przez trzydzieści lat pracował w Computing Science Research Center of Bell Laboratories. Jego zainteresowania badawcze obejmują narzędzia programowe, języki zorientowane na aplikacje, metodologię programowania, interfejsy użytkownika, humanistykę cyfrową i edukację technologiczną. Jest członkiem American Academy of Arts and Sciences i National Academy of Engineering.

Helion

helion.pl

HELION SA
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 98 63
helion@helion.pl



ISBN 978-83-283-7163-7



Cena: 49,00 zł