Mainframe i z/OS, część 9 z 9: Epilog

Epilog

Mainframe to maszyna, która wygrała wszystko, co dało się wygrać, i przez pięćdziesiąt lat była z tego powodu nieustannie uśmiercana.

Wygrał zakład o pięć miliardów. Ustalił, że bajt ma osiem bitów. Wynalazł wirtualizację czterdzieści lat przed resztą świata i klastrowanie ze współdzielonym zapisem dwadzieścia lat przed resztą świata. Urodził SQL — i oddał go konkurencji, bo bał się skanibalizować własny produkt. Przetrwał trzynastoletni proces antymonopolowy, japońskie szpiegostwo przemysłowe, radzieckie klonowanie, rewolucję klient-serwer, Linuksa, chmurę i cztery fale przepowiedni o własnej śmierci.

Jego historia jest przestrogą i pociechą naraz — dokładnie jak historia każdej dobrze zaprojektowanej rzeczy.

Przestrogą, bo pokazuje, jak drogo kosztuje architektoniczna decyzja podjęta bez marginesu. Dwadzieścia cztery bity adresu w 1964 roku wyglądały na ekstrawagancję; osiemnaście lat później kosztowały najbardziej bolesną operację w dziejach tej platformy, a ich ślady — „above the line”, AMODE, RMODE — siedzą w narzędziach do dziś. Pokazuje też, że firma może wynaleźć przyszłość i jej nie sprzedać: IBM miał Codda, System R i SQL, a na rynek wyszedł jako trzeci, bo bał się o własny dochodowy produkt.

Pociechą, bo dowodzi, że *dobrze zaprojektowana rzecz ma niebywałą siłę przetrwania*. Program skompilowany w 1966 roku uruchomi się dziś. Nie przez sentyment, tylko dlatego, że ktoś kiedyś potraktował obietnicę zgodności śmiertelnie poważnie i nie złamał jej ani razu przez sześć dekad — mimo czterech zmian technologii krzemowej, trzech zmian szerokości adresu i kilkunastu zmian strategii korporacyjnej. W branży, w której framework webowy przestaje być wspierany po trzech latach, jest w tym coś niemal wzruszającego.

I jest jeszcze trzecia lekcja, najbardziej praktyczna. Mainframe nie przetrwał dlatego, że był najlepszy technicznie — w wielu wymiarach nie był. Przetrwał, bo **koszt zastąpienia go zawsze był wyższy niż koszt jego utrzymania**. To jest ta sama siła, która trzyma przy życiu każdy system, o którym ktoś w twojej firmie mówi „kiedyś to przepiszemy”. Warto o niej pamiętać, projektując cokolwiek, co ma żyć dłużej niż jeden kwartał — bo prawdopodobieństwo, że twój kod przeżyje twoje plany wobec niego, jest znacznie wyższe, niż się wydaje.

Szesnaście systemów, jedna lekcja

Zestawienie obejmuje całą serię i jest utrzymywane w jednym miejscu — na stronie *Tabela systemów i porównanie poleceń*.

Do tego dochodzi **POSIX**, który nie jest systemem i dlatego nie ma go w tabeli — a jest jedyną rzeczą spinającą kilka jej wierszy: dokumentem, dzięki któremu kod napisany dla jednego z nich działa na pozostałych.

Pięć wniosków, których żadna z tych historii nie pokazuje osobno.

Pierwszy: jakość techniczna nie wystarcza, a czasem bywa zbędna. OpenVMS miał klastry dwadzieścia lat przed konkurencją i zginął. BSD było lepsze od Linuksa o dekadę i przegrało. OS/2 był lepszy od Windows 3.1 pod każdym względem technicznym i przegrał. CP/M był jedynym poważnym systemem dla mikrokomputerów i przegrał z kopią napisaną w kilka tygodni. A DOS i Windows 3.x nie miały ani jednej przewagi technicznej i wygrały wszystko. To jest niewygodna prawda o tej branży, którą lepiej nazwać wprost, niż udawać, że inżynieria obroni się sama.

Drugi: o losie platformy decyduje ekonomia jej ekosystemu, nie architektura jądra. Mainframe żyje, bo migracja kosztuje więcej niż utrzymanie. UNIX rozpadł się, bo każdy producent chciał własnej wersji. OS/2 przegrał, bo tak dobrze uruchamiał cudze programy, że nikt nie pisał dla niego własnych. CP/M przegrał na różnicy cen czterdziestu i dwustu czterdziestu dolarów. DOS wygrał dzięki jednemu zdaniu w umowie z 1980 roku. Windows 3.x wygrał, bo działał na sprzęcie, który już stał na biurku. BSD przegrało dwa lata w sądzie. Linux wygrał, bo nikt nie musiał się zastanawiać, czy wolno. **Osiem różnych mechanizmów — i ani jeden z nich nie jest kwestią techniczną.**

Trzeci: niepewność jest droższa niż zakaz. Nikt nie zakazał BSD — po prostu przez dwa lata nie było wiadomo, czy wolno. Nikt nie zakazał OS/2 — nie było wiadomo, czy przetrwa. SCO nikomu niczego nie udowodniło, ale przez trzy lata wystarczyła sama groźba. Kod AARD nie blokował niczego — wystarczyło, że wyświetlał niejasny komunikat i pozwalał podejrzewać. **W decyzjach technologicznych brak odpowiedzi kosztuje więcej niż odpowiedź odmowna**, bo z zakazem da się coś zrobić, a z niewiedzą nie.

Czwarty: zgodność częściowa bije zgodność pełną. Copland próbował dać pełną zgodność wstecz i nie dał niczego; Carbon zaakceptował, że przeniesie siedemdziesiąt procent, i przeniósł cały ekosystem. Windows 3.x nie próbował uruchamiać programów DOS-owych *doskonale* — uruchamiał je *wystarczająco dobrze*, w oknie, i to wystarczyło, by ludzie przesiedli się stopniowo zamiast jednym skokiem. OS/2 poszedł drogą doskonałości i przegrał.

Piąty, najbardziej pocieszający: dobrze pomyślana rzecz przeżywa własną śmierć. Każdy system z tej tabeli został w pewnym momencie formalnie uśmiercony przez firmę, która go stworzyła — i prawie każdy dostał w ostatnich latach nowe wydanie. OpenVMS ma port na x86-64.

Mainframe ma akcelerator sztucznej inteligencji w procesorze. OS/2 startuje z UEFI. DOS dostał wydanie 1.4 od ochotników. NeXT, firma uznana za porażkę, stoi pod dwoma i pół miliardami urządzeń. A CP/M — system, który przegrał wszystko i którego producenta nie ma od trzydziestu pięciu lat — zostawił po sobie litery dysków, nazwy plików 8.3, kropkę jako separator rozszerzenia i warstwę abstrakcji sprzętu, której używasz codziennie, nie wiedząc o tym.

> *„Ostatni mainframe zostanie odłączony od prądu 15 marca 1996 roku.”*
> — Stewart Alsop, *InfoWorld*, 1991

Trzydzieści lat po tej dacie IBM wypuścił z17.

Ciekawostki zebrane

- **PDP i IBM z tego samego powodu.** DEC nazywał swoje maszyny „procesorami danych”, bo inwestorzy zabronili mu budować komputery. IBM zmienił nazwę z CTR, bo Watson uznał poprzednią za niezgrabną. Dwie najważniejsze marki tej branży nazywają się tak, jak się nazywają, z powodów czysto pozatechnicznych.
- **Osiem bitów, bo jeden inżynier wygrał spór.** Ośmiobitowy bajt jest standardem świata, bo Gerrit Blaauw przeforsował go w 1963 roku wobec ówczesnego standardu sześciu bitów i presji kosztowej, żeby zostały cztery.
- **Bit, który nigdy nic nie robił.** Architektura System/360 ma w PSW bit przełączania w tryb ASCII. Całe oprogramowanie systemowe napisano pod EBCDIC, więc bit istniał i nie znaczył nic.
- **Alfabet z dziurami.** W EBCDIC litery A–I, J–R i S–Z leżą w trzech osobnych zakresach, bo odpowiadają trzem kombinacjom stref na karcie dziurkowanej. Pętla „od A do Z” napotyka 41 pozycji kodowych zamiast 26.
- **Drewniane peryferia.** Część urządzeń pokazanych dziennikarzom 7 kwietnia 1964 roku to były atrapy ze sklejki.
- **Program, który nic nie robi i jest wszędzie.** `IEFBR14` składa się z jednej instrukcji powrotu. Jest jednym z najczęściej uruchamianych programów w historii, bo służy do alokowania zbiorów przez sam fakt przetworzenia kart JCL.
- **Formatowanie czeku w jednej instrukcji.** Instrukcje `ED` i `EDMK` tłumią zera wiodące, wstawiają separatory tysięcy i przecinek dziesiętny, obsługują znak i wypełniacz — a `EDMK` zwraca jeszcze adres pierwszej znaczącej cyfry, żeby wstawić przed nią symbol waluty. Jedna instrukcja procesora, rok 1964.
- **Terminal, który nie wysyła znaków.** Dzięki trybowi blokowemu 3270 jeden IBM 3033 z 16 megabajtami pamięci obsługiwał pod CICS-em do **17 500 terminali**.
- **Serverless z 1969 roku.** Model pseudokonwersacyjny CICS-a — transakcja kończy się z wysłaniem ekranu, stan wędruje w jawnie przekazywanej strukturze — to jest bezstanowa obsługa żądań wynaleziona pół wieku przed REST-em, z powodu ceny pamięci.
- **Pomysł w autobusie.** Robert Creasy wpadł na koncepcję CP-40, czyli maszyn wirtualnych, jadąc autobusem MTA z Arlington do Cambridge pod koniec grudnia 1964

roku.

- **Podręcznik jako specyfikacja interfejsu.** Twórcy CP-40 wzięli *System/360 Principles of Operation* jako opis tego, co ma widzieć użytkownik. Stąd wierność maszyny wirtualnej — i stąd to, że **CP-67 mógł działać pod CP-67**.
- **Emulator emulatora.** CP-40 działał najpierw jako symulacja sprzętu Modelu 67 uruchomiona na Modelu 40 — zanim prawdziwy Model 67 powstał.
- **Prawie czterdzieści lat.** IBM miał czystą, pełną wirtualizację w 1967 roku — Model 67 z systemem CP-67. Architektura x86 dostała sprzętowe wsparcie klasycznej wirtualizacji dopiero w latach 2005–2006 (VT-x 14 XI 2005, AMD-V 2006); czy wcześniej spełniała kryteria Popka i Goldberga, źródła oceniają różnie[1].
- **Hipernadzorca nie stąd.** Słowo *hypervisor* IBM ukuł nie dla CP-67, tylko dla oprogramowania zarządzającego równoczesnym działaniem OS/360 i emulatora 7090 na maszynie 360/65. Podział na typ 1 i typ 2 pochodzi z doktoratu Roberta Goldberga z 1973 roku.
- **Cray odpowiada Watsonowi.** Na notatkę o laboratorium CDC liczącym „34 osoby, wliczając woźnego” Seymour Cray odpowiedział: *„It seems like Mr. Watson has answered his own question.”*
- **Komputer bez asemblera — dla klienta.** Burroughs B5000, opisany w 1961 roku, nie dawał użytkownikom asemblera, a MCP od B5500 pisano w ESPOL, rozszerzonym ALGOL-u. Pierwszy, bębnowy MCP powstał jednak w asemblerze OSIL na Burroughsie 220[4]. Bit flagi w pamięci odróżniał dane od słów sterujących, ale ochrona zależała też od zaufanych kompilatorów[3].
- **Wyrok, który oddał komputer ludzkości.** 19 października 1973 sąd unieważnił patent na ENIAC, orzekając, że elektroniczny komputer cyfrowy wynalazł John Vincent Atanasoff. Wynalazek trafił do domeny publicznej.
- **Chłodzenie powietrzem jako argument sprzedażowy.** Amdahl wygrywał z IBM między innymi tym, że jego maszyny nie wymagały instalacji chłodzenia wodą.
- **Grzywna 10 000 dolarów.** Tyle wyniosła maksymalna kara dla Hitachi za szpiegostwo przemysłowe w sprawie IBM 3081. Wszystkie kary razem: 24 tysiące. Za informacje zapłacono agentom FBI od 495 do 650 tysięcy[6].
- **Wietnam wydziału antymonopolowego.** Tak Robert Bork nazwał sprawę USA v. IBM: trzynaście lat, 950 świadków, 30 milionów stron dokumentów, umorzenie „bez podstaw”.
- **Dzień narodzin branży.** 23 czerwca 1969 IBM rozdzielił cenniki sprzętu i oprogramowania. Powszechnie uważa się, że tym samym stworzył przemysł oprogramowania.
- **IBM 360/40 w walizkach.** Krąży relacja, że pierwszy radziecki mainframe zgodny z IBM powstał na podstawie egzemplarza przemyconego *przez Polskę* — rozmontowanego i przewiezionego w dużej liczbie walizek. Dokumentu brak, a wersji są trzy[10].
- **Najwydajniejsza w Brnie.** Wrocławska R-32 z Elwro wypadła najlepiej w porównaniu maszyn Jednolitego Systemu w Brnie w 1973 roku.
- **Odra i ICT: to była umowa.** Kompatybilność Odry 1300 z brytyjską serią ICL 1900 wzięła się z porozumienia z firmą ICT, podpisanego w lipcu 1967 roku, a nie z inżynierii wstecznej.
- **30 kwietnia 2010, godzina 22:00.** Wtedy w Lublinie wyłączono ostatnie Odry 1305, na których działał kolejowy system kierowania pracą stacji rozrządowych — system, który pracował trzy zmiany przez trzydzieści cztery lata.

- **Gigaherce, które się cofnęły.** zEC12 z 2012 roku taktowany był 5,5 GHz — szybciej niż cztery następne generacje. IBM świadomie obniżył zegar na rzecz szerszego wykonania. Dopiero z17 (2025) wrócił do 5,5 GHz.
- **Cache, który jest cudzy.** W procesorze Telum nie ma dedykowanego L3 ani L4 — funkcję wirtualnego L3 pełnią cache'e L2 sąsiednich rdzeni, dając 2 GB „L4” na szufladę bez osobnego krzemu.
- **Dysk, którego nie ma.** z/OS adresuje dane przez numer cylindra i głowicy nieistniejącego talerza magnetycznego, a macierz dyskowa syntetyzuje tę geometrię na dyskach SSD.
- **Siedem dziewiątek z ankiety.** Słynna dostępność 99,99999% pochodzi z corocznego badania, w którym firmy *same deklarują* swoją dostępność w formularzu internetowym — nie z pomiaru MTBF. Tę samą liczbę podawało w 2005 roku HP w reklamie serwerów NonStop; o innej drodze do odporności — niezależnych procesorach bez współdzielenia i parach procesów zamiast lockstepu — oraz o pomiarach przestojów Jima Graya z 1985 roku pisze strona **Tandem NonStop**.
- **Płacisz za to, żeby nie pracować za szybko.** Szczytowa wartość kroczącej czterogodzinnej średniej zużycia mocy wyznacza rachunek za cały miesiąc — dlatego administratorzy sztucznie ograniczają moc maszyny, za którą zapłacili.
- **Graffiti za grzywnę.** IBM promował Linuksa na mainframe'ach, malując pingwiny na chodnikach Chicago i San Francisco. Miasta nałożyły kary.
- **Alsop zjadł swoje słowa dosłownie.** Zdjęcie z 2002 roku jest eksponatem Computer History Museum.

Revision #1
Created 2026-11-03 00:55:05 UTC by Tomasz Konieczny
Updated 2026-11-03 00:55:05 UTC by Tomasz Konieczny