

Nowa strona

DOS — kronika systemu, który zbudował imperium z przypadku

Historia operacyjnego systemu napisanego „na szybko i byle jak”, kupionego za grosze, który dzięki jednej genialnej klauzuli licencyjnej opanował świat, przez piętnaście lat był fundamentem każdego peceta — a potem został zdegradowany do roli programu rozruchowego i ostatecznie wyparty przez linię Windows NT. I który, mimo to, wciąż żyje.

“To być może najlepszy w dziejach informatyki dowód, że o sukcesie technologii decyduje nie jej jakość, lecz **umowa, którą się wokół niej podpisze**. DOS nie był najlepszy. Nie był nawet oryginalny. Ale stał za nim kontrakt, który zamienił system warty 75 tysięcy dolarów w fundament imperium wartego setki miliardów.

Spis treści

1. Wprowadzenie: system, którego prawie nie było
2. Prehistoria: świat przed DOS-em
3. Projekt Chess: jak IBM w pośpiechu zbudował peceta
4. Feralne spotkanie: IBM, Digital Research i legenda Kildalla
5. Quick and Dirty: narodziny 86-DOS
6. Deal stulecia: Microsoft kupuje DOS-a
7. 12 sierpnia 1981: IBM PC i PC DOS 1.0
8. Anatomia DOS-u: jak to właściwie działało
9. Dojrzewanie: wersje 1.x-3.x

10. Klony i imperium licencji
 11. Bariera 640 kB i wojny o pamięć
 12. Wersje 4.0 i 5.0: potknięcie i odkupienie
 13. Wojna z Digital Research: DR-DOS, FUD i kod AARD
 14. Wersje 6.x i wojna o kompresję: sprawa Stac
 15. Windows na grzbiecie DOS-u
 16. Zmierzch: linia NT pożera DOS-a
 17. Aplikacje, które trzymały DOS przy życiu
 18. Złoty wiek gier DOS
 19. Programowanie pod DOS-em
 20. Skrypty wsadowe: programowanie dla każdego
 21. Życie po śmierci: FreeDOS i wieczność
 22. Epilog: dlaczego DOS wciąż jest wszędzie
 23. Ciekawostki zebrane
 24. Oś czasu
 25. Źródła i dalsze lektury
-

1. Wprowadzenie: system, którego prawie nie było

VMS przegrał przez korporacyjne fuzje. OS/2 przegrał przez arogancję i marketing. DOS jest trzecią stroną tej samej monety — z tą różnicą, że DOS **wygrał**, i to wygrał miażdżąco, mimo że pod względem czysto technicznym był najuboższy z całej trójki. To paradoks, który czyni jego historię tak pouczającą: gdy VMS i OS/2 udowadniają, że doskonałość nie wystarcza, DOS dowodzi rzeczy jeszcze bardziej niepokojącej — że **doskonałość bywa wręcz zbędna**, jeśli ma się właściwą umowę i właściwy moment.

DOS był systemem jednozadaniowym, jednoużytkownikowym, bez ochrony pamięci, bez prawdziwej wielozadaniowości, uwięzionym w architekturze procesora z lat 70. Był — dosłownie, zgodnie z pierwotnym akronimem — „szybki i byle jaki”. A jednak przez kilkanaście lat działał na setkach milionów komputerów, dał Microsoftowi paliwo do zbudowania monopolu i ukształtował sposób, w jaki cały świat myśli o komputerze osobistym: litery dysków, ukośniki wsteczne w ścieżkach, rozszerzenia plików, pliki `CONFIG.SYS` i `AUTOEXEC.BAT`. Echa tych konwencji żyją w Windowsie do dziś.

Ta opowieść prowadzi przez całą trajektorię: od świata przed DOS-em, przez genialny w swej bezwzględności kontrakt z IBM, przez złoty wiek i barokowe wojny o każdy kilobajt pamięci, aż po powolne wchłonięcie przez Windows i ostateczne wyparcie przez linię NT. A na końcu — jak zwykle w tej serii — dowiemy się, że system formalnie pogrzebany w 2000 roku wcale nie umarł.

2. Prehistoria: świat przed DOS-em

Aby zrozumieć DOS-a, trzeba poznać system, którego był on bladą kopią: **CP/M**.

W połowie lat 70., gdy mikrokomputery dopiero raczkowały, niejaki **Gary Kildall** — błyskotliwy informatyk, doktor, wykładowca i konsultant Intela — napisał system operacyjny dla 8-bitowych procesorów Intel 8080. Nazwał go **CP/M** (*Control Program for Microcomputers*) i zaczął sprzedawać przez założoną przez siebie firmę **Digital Research, Inc. (DRI)**. Geniusz CP/M polegał na **warstwie abstrakcji sprzętu** (BIOS), dzięki której ten sam system i te same programy działały na maszynach różnych producentów — wystarczyło dopasować cienką warstwę sterującą. W świecie setek niezgodnych ze sobą konstrukcji opartych na magistrali **S-100** było to zbawienie.

Pod koniec lat 70. CP/M był **de facto standardem** dla poważnych mikrokomputerów — królem, którego pozycja wydawała się niezagrożona. Powstała wokół niego ogromna biblioteka oprogramowania (m.in. legendarny edytor WordStar i arkusz SuperCalc). Gdyby historia potoczyła się inaczej, dziś mówilibyśmy o „komputerze CP/M” tak, jak mówimy o „pececie”. Konwencje, które dziś kojarzymy z DOS-em — litery dysków, nazwy plików w formacie 8.3, polecenia takie jak `DIR` czy `TYPE` — w ogromnej mierze pochodziły właśnie z CP/M. DOS był, mówiąc wprost, jego naśladowcą.

“**Ciekawostka.** To, że DOS tak bardzo przypominał CP/M, nie było przypadkiem — jego autor pisał go z podręcznikami CP/M na biurku, celowo kopiując wywołania API, by ułatwić przenoszenie programów. Ta bliskość stanie się później źródłem wieloletnich oskarżeń o plagiat, których echo słychać do dziś.

3. Projekt Chess: jak IBM w pośpiechu zbudował peceta

Na przełomie lat 70. i 80. IBM — gigant mainframe'ów i minikomputerów — patrzył na rynek komputerów osobistych z boku, z lekceważeniem. Ale sukces Apple II, TRS-80 i Commodore robił się zbyt głośny, by go ignorować. Pod koniec 1980 roku zapadła decyzja: IBM zbuduje własnego peceta.

Problem w tym, że tradycyjny cykl projektowy IBM trwał około czterech lat — w świecie mikrokomputerów wieczność. Rozwiązaniem był **Projekt Chess**: niewielki, zwinny zespół ulokowany w **Boca Raton na Florydzie** (Entry Systems Division), z bezprecedensowym mandatem do **obejścia korporacyjnych procedur** i dostarczenia gotowego produktu w **rok**.

Zespół, początkowo dwunastoosobowy i ochrzczony „Dirty Dozen” („parszywa dwunastka”), prowadził najpierw William C. Lowe, a potem **Don Estridge**.

Presja czasu wymusiła decyzję, która zmieni dzieje informatyki. Zamiast projektować wszystko od zera w stylu IBM, zespół sięgnął po **gotowe komponenty z rynku**: procesor Intel 8088, układy peryferyjne innych firm, standardowe części. A architekturę uczyniono **otwartą** — z opublikowaną dokumentacją techniczną i gniazdami rozszerzeń dla sprzętu firm trzecich. To otwarcie, podyktowane pośpiechem (i być może przekonaniem części kadry, że pecet nie ma wielkiej przyszłości), okaże się zarazem największym atutem i największym przekleństwem IBM: pozwoli zbudować standard, ale i odda go w ręce klonów.

Jednego elementu IBM nie miał: **systemu operacyjnego**. I po niego wybrał się na zewnątrz.

4. Feralne spotkanie: IBM, Digital Research i legenda Kildalla

Logika podpowiadała jedno: skoro CP/M jest standardem, nowy pecet IBM powinien działać pod CP/M. Latem 1980 roku przedstawiciele IBM ruszyli więc do Digital Research, by licencjonować szykowaną właśnie 16-bitową wersję systemu, **CP/M-86**.

I tu rodzi się jedna z najśłynniejszych legend Doliny Krzemowej — opowieść o tym, jak Gary Kildall „przegapił” spotkanie z IBM, bo wolał latać swoim samolotem. Rzeczywistość była bardziej prozaiczna i bardziej ludzka. Negocjacje po stronie DRI prowadziła m.in. **Dorothy McEwen-Kildall** (żona Gary'ego i szefowa spraw licencyjnych), która zawahała się przed podpisaniem rygorystycznej **umowy o poufności (NDA)** przedłożonej przez IBM. Kością niezgody okazał się jednak przede wszystkim **model rozliczeń**: IBM chciał wykupić system za jednorazową, ryczałtową kwotę (mówi się o 250 000 dolarów) za nieograniczoną liczbę kopii, podczas gdy Digital Research obstawało przy klasycznym modelu **tantem od każdej sprzedanej sztuki**. Strony się nie dogadały, a IBM odszedł z kwitkiem.

Legenda o „lataniu zamiast spotkania” jest krzywdząca i w dużej mierze nieprawdziwa — ale dobrze oddaje gorzki morał: **Digital Research miało w ręku rynek i je wypuściło**, nie z powodu jednego nieobecnego dnia, lecz z powodu nieufności i nieugiętości w negocjacjach. Ta sama nieugiętość, która w innych okolicznościach byłaby cnotą, tu okazała się zgubna.

IBM wrócił więc do firmy, z którą już się dogadał w sprawie języka BASIC — do młodego, głodnego Microsoftu.

5. Quick and Dirty: narodziny 86-DOS

Microsoft w 1980 roku był przede wszystkim dostawcą interpreterów języka BASIC. **Systemu operacyjnego nie miał żadnego.** Gdy więc IBM — być może w przekonaniu, że Microsoft dysponuje CP/M (bo sprzedawał kartę SoftCard pozwalającą uruchamiać CP/M na Apple II) — poprosił Billa Gatesa także o system, Gates musiał coś szybko znaleźć.

Szczęście (lub spryt) sprawiło, że odpowiedni produkt leżał tuż za rogiem. Niewielka firma **Seattle Computer Products (SCP)** z Tukwili w stanie Waszyngton produkowała 16-bitowe płyty z procesorem Intel 8086 — ale nie miała dla nich systemu, bo zapowiadane przez Digital Research CP/M-86 wciąż się opóźniało. Aby ratować sprzedaż, programista SCP **Tim Paterson** usiadł w kwietniu 1980 roku i przez około cztery miesiące napisał własny system od podstaw, w assemblerze, z podręcznikami CP/M pod ręką. Nazwał go bez ogródek **QDOS** — *Quick and Dirty Operating System*, „szybki i byle jaki system operacyjny”.

QDOS celowo kopiował wywołania API z CP/M, by ułatwić przenoszenie istniejących programów. Wersję 0.11 ukończono w sierpniu 1980, sprzedaż ruszyła we wrześniu, a w grudniu 1980 produkt przemianowano na bardziej godne **86-DOS**.

“ **Ciekawostka.** Cały fundament późniejszego imperium Microsoftu — system, który trafi na setki milionów maszyn — powstał jako *pro wizorka* jednego programisty, ratująca sprzedaż płyt głównych niewielkiej firmy. Nazwa „Quick and Dirty” nie była ironią. Była opisem.

6. Deal stulecia: Microsoft kupuje DOS-a

Tu zaczyna się najważniejsza część całej historii — bo to nie kod, lecz **umowa** zbudowała imperium.

W grudniu 1980 roku Microsoft wykupił od SCP **niewyłączną licencję** na 86-DOS za **25 000 dolarów**. W maju 1981 zatrudnił samego Tima Patersona, by przeniósł system na procesor Intel 8088 stosowany w nadchodzącym IBM PC. A **27 lipca 1981 roku**, na kilka tygodni przed premierą peceta, Microsoft odkupił od SCP **pełne prawa** do systemu za dodatkowe **50 000 dolarów** — łącznie więc około **75 000 dolarów**. 86-DOS przemianowano na **MS-DOS**, a wersję dla IBM

nazwano **PC DOS**.

I teraz klauzula, która zmieniła wszystko. Microsoft tak ułożył umowę z IBM, że **zachował prawo do licencjonowania MS-DOS innym producentom**. IBM dostał system do swojego peceta — ale nie dostał go na wyłączność. W 1980 roku, gdy pecet IBM był pojedynczym, jeszcze nieistniejącym produktem, wyglądało to na drobiazg. W rzeczywistości była to **najlepsza decyzja biznesowa w historii oprogramowania**. Gdy tylko pojawią się klony peceta IBM — a pojawią się szybko — każdy z nich będzie potrzebował systemu. I każdy kupi go od Microsoftu.

“ **Ciekawostka — i gorzki epilog dla SCP.** Seattle Computer Products dostało perpetualną, wolną od opłat licencję na sprzedaż 86-DOS ze swoim sprzętem — ale nie miało pojęcia, że klientem Microsoftu jest IBM. Gdy prawda wyszła na jaw, SCP pozwało Microsoft o oszukańcze zatajenie tej informacji. Sprawa zakończyła się ugodą. Firma, która stworzyła przyszłego króla systemów operacyjnych, sprzedała go w całości za 75 tysięcy dolarów — i zniknęła z historii.

7. 12 sierpnia 1981: IBM PC i PC DOS 1.0

12 sierpnia 1981 roku w nowojorskim hotelu Waldorf-Astoria IBM zaprezentował **IBM Personal Computer model 5150**: procesor Intel 8088 taktowany 4,77 MHz, od 16 do 64 kB RAM, napęd dyskietek 5,25 cala i system **PC DOS 1.0** w cenie zaczynającej się od około 1565 dolarów. Choć architektonicznie był to w dużej mierze ulepszony 86-DOS, IBM poddał go rygorystycznej kontroli jakości — podobno wykryto i poprawiono ponad 300 błędów.

Klient peceta miał teoretycznie wybór systemu: IBM oferował zarówno PC DOS, jak i — z opóźnieniem — CP/M-86 Gary'ego Kildalla, który dostrzegł podobieństwo DOS-u do swojego dzieła i zagroził IBM procesem (jego prawnik odradził jednak pozew). Wybór rozstrzygnęła jednak nie technika, lecz cena: IBM wycenił CP/M-86 na **240 dolarów**, a PC DOS na **40 dolarów**. Przy sześciokrotnej różnicy ceny i niemal identycznej funkcjonalności wynik był przesądzony. CP/M-86 zniknął, a DOS — kopia CP/M — zajął jego tron. Ironia była doskonała.

The IBM Personal Computer DOS

Version 1.00 (C)Copyright IBM Corp 1981

A>DIR

COMMAND COM 3231 8-04-81 12:00p

```
FORMAT COM 2560 8-04-81 12:00p
DISKCOPY COM 1408 8-04-81 12:00p
3 File(s)
```

```
A>_
```

Sukces peceta IBM przerósł wszelkie oczekiwania samego IBM. A wraz z nim w górę poszybował los systemu, który Microsoft kupił za grosze — bo z każdym sprzedanym pecetem (i, wkrótce, z każdym klonem) sprzedawała się kolejna kopia DOS-u.

8. Anatomia DOS-u: jak to właściwie działało

DOS był prosty — i ta prostota była zarazem jego zaletą (działał na skromnym sprzęcie) i ograniczeniem (nie chronił przed niczym). Przyjrzyjmy się, z czego się składał.

Tryb rzeczywisty i brak ochrony

DOS działał w **trybie rzeczywistym** procesora x86: każdy program miał pełny, niczym nieograniczony dostęp do całej pamięci i sprzętu. Nie było ochrony pamięci ani izolacji procesów — błędny lub złośliwy program mógł nadpisać pamięć systemu i zawiesić maszynę. DOS był też z gruntu **jednozadaniowy**: uruchamiał jeden program, oddawał mu kontrolę i czekał, aż ten skończy.

Trzy filary systemu

System tworzyły zasadniczo trzy pliki: dwa ukryte moduły jądra (u IBM nazwane `IBMBIO.COM` i `IBMDOS.COM`, u Microsoftu `IO.SYS` i `MSDOS.SYS`) oraz interpreter poleceń `COMMAND.COM` — to z nim rozmawiał użytkownik, wpisując polecenia przy znaku zachęty. Polecenia dzieliły się na **wewnętrzne** (wbudowane w `COMMAND.COM`, jak `DIR`, `COPY`, `TYPE`, `CD`) i **zewnętrzne** (osobne programy, jak `FORMAT.COM` czy `CHKDSK.EXE`). Słynny komunikat `Bad command or file name` pojawiał się, gdy interpreter nie znalazł ani polecenia, ani pliku o danej nazwie.

INT 21h: brama do systemu

Programy korzystały z usług DOS-u przez **przerwanie programowe 21h** (`INT 21h`) — odpowiednik dzisiejszego interfejsu wywołań systemowych. Ustawiało się numer funkcji w rejestrze i wywoływało

przerwanie: otwarcie pliku, odczyt, zapis, alokacja pamięci, wypisanie znaku. Cała komunikacja aplikacji z systemem przechodziła przez tę jedną, kultową bramę.

FAT i format 8.3

Dane organizował system plików **FAT** (*File Allocation Table*) — prosta tablica śledząca, które klastry dysku należą do których plików. Nazwy plików miały sztywny format **8.3**: do ośmiu znaków nazwy, kropka i do trzech znaków rozszerzenia (`RAPORT.TXT`, `COMMAND.COM`). Programy wykonywalne miały rozszerzenia `.COM` (proste, do 64 kB), `.EXE` (większe, relokowalne) i `.BAT` (skrypty wsadowe). Litery dysków (`A:`, `B:` dla dyskietek, `C:` dla pierwszego twardego) i ukośnik wsteczny w ścieżkach to spuścizna, którą Windows nosi do dziś.

CONFIG.SYS, AUTOEXEC.BAT i TSR-y

Przy starcie DOS czytał dwa pliki konfiguracyjne. `CONFIG.SYS` ładował sterowniki urządzeń i ustawiał parametry jądra; `AUTOEXEC.BAT` był skryptem wsadowym wykonywanym automatycznie po starcie (ustawienie ścieżki, uruchomienie programów). Ponieważ DOS nie umiał działać wielozadaniowo, sprytni programiści wymyślili **TSR-y** (*Terminate and Stay Resident*) — programy, które „kończyły działanie”, ale zostawały w pamięci, reagując na przerwania (np. wyskakujący kalendarzyk po wciśnięciu klawisza). To była namiastka wielozadaniowości doklejona do systemu, który jej z założenia nie miał.

```
REM ===== Przykładowy CONFIG.SYS =====
```

```
DEVICE=C:\DOS\HIMEM.SYS
```

```
DEVICE=C:\DOS\EMM386.EXE NOEMS
```

```
DOS=HIGH,UMB
```

```
FILES=30
```

```
BUFFERS=20
```

```
DEVICEHIGH=C:\DOS\ANSI.SYS
```

```
SHELL=C:\COMMAND.COM /P /E:512
```

```
REM ===== Przykładowy AUTOEXEC.BAT =====
```

```
@ECHO OFF
```

```
PATH C:\DOS;C:\UTILS
```

```
PROMPT $P$G
```

```
SET TEMP=C:\TEMP
```

```
LOADHIGH C:\MOUSE\MOUSE.COM
```

```
ECHO Witaj w systemie!
```

☐ **Ciekawostka.** Microsoft świadomie nie dodał do DOS-u wielu użytkowników ani wielozadaniowości, bo... miał już od tego osobny system: uniksowy **Xenix**. Plan był taki, by DOS pozostał prostym, jednoużytkowym systemem, a po moc sięgać do Xeniksa. Gdy jednak po rozpadzie Bell System rynek Uniksa zdominował AT&T, Microsoft uznał, że nie wygra tej bitwy, i w 1987 roku oddał Xeniksa firmie SCO. DOS został sam — i to on, nie Xenix, podbił świat.

9. Dojrzewanie: wersje 1.x–3.x

DOS dojrzewał skokami, wersja po wersji, często podążając za nowym sprzętem IBM.

DOS 1.0 (1981) był prymitywny: obsługiwał tylko jednostronne dyskietki 160 kB, nie znał podkatalogów (wszystkie pliki w katalogu głównym), nie miał przekierowań ani sterowników instalowalnych. **DOS 1.25** (1982) był pierwszą wersją licencjonowaną producentom **spoza IBM** — i to ona dała początek całemu ekosystemowi OEM-owych odmian (Compaq-DOS, Zenith Z-DOS, TI itd.).

Prawdziwym przełomem był **DOS 2.0** (marzec 1983), wydany wraz z **IBM PC XT** i jego twardym dyskiem 10 MB. To w zasadzie przeprojektowanie systemu: wprowadzono **podkatalogi** (hierarchiczne drzewo katalogów, wzorowane na Uniksie), obsługę **dysków twardych**, **instalowalne sterowniki urządzeń** (przez `CONFIG.SYS`), a także uniksowe **przekierowania i potoki** (`>`, `<`, `|`). DOS 2.0 z jednozabawki dla dyskietek stał się systemem zdolnym do poważnej pracy.

Linia **3.x** dograła resztę. **DOS 3.0** (1984) przyszedł z **IBM PC AT**, wsparciem dla FAT16 (partycje do 32 MB) i dyskietek 1,2 MB. **DOS 3.1** (1985) dodał obsługę **sieci lokalnych** (Microsoft Networks). **DOS 3.2** (1986) — dyskietki 3,5 cala 720 kB. **DOS 3.3** (1987), wydany z linią PS/2, dorzucił dyskietki 1,44 MB i partycje rozszerzone. Wersja 3.3 uchodzi do dziś za jedną z najbardziej lubianych i stabilnych — prawdziwego „konia roboczego” epoki.

10. Klony i imperium licencji

Teraz genialna klauzula z rozdziału 6 zaczęła przynosić owoce — i to jakie.

Otwarta architektura peceta IBM oznaczała, że niemal każdy mógł zbudować zgodną maszynę z tych samych, ogólnodostępnych części. Jedyną przeszkodą był **BIOS** — firmware IBM chroniony prawem autorskim. Ale i tę barierę szybko obeszli inżynierowie, stosując **czysty pokój** (*clean-room reverse engineering*): jeden zespół opisywał, co BIOS robi, drugi — nie widząc oryginalnego kodu — pisał własny od zera. Tak powstały legalne, zgodne BIOS-y firm **Phoenix**, **American Megatrends (AMI)** i **Award**. Pierwszy funkcjonalny klon (Columbia Data Products) pojawił się już w 1982, a

Compaq Portable z 1983 roku reklamowano jako „w stu procentach zgodny”.

Ruszyła lawina. Rynek zalały setki producentów „kompatybilnych z IBM PC” — często tańszych i lepszych niż oryginał. I każda z tych maszyn potrzebowała systemu operacyjnego. Dzięki klauzuli o niewyłącznym licencjonowaniu wszystkie kupowały go od jednej firmy: **Microsoftu**. IBM stworzył standard, ale to Microsoft zgarniał haracz od całego standardu. Pecet stał się towarem masowym, a MS-DOS — jego niewidzialnym, obowiązkowym podatkiem.

Model licencyjny Microsoftu był przy tym przemysłnie skonstruowany. Producentom oferowano umowy **per-procesor** (opłata od każdej wyprodukowanej maszyny, niezależnie od zainstalowanego systemu), per-system lub per-kopia. Najwięksi wybierali wariant per-procesor, bo miał najniższą stawkę — ale to oznaczało, że **płacili Microsoftowi za DOS także wtedy, gdy chcieli zainstalować coś innego**. Dołożenie konkurencyjnego systemu (jak DR-DOS) stawało się ekonomicznym absurdem. Ten mechanizm, błyskotliwie skuteczny i — jak się okaże — prawnie wątpliwy, zabetonował dominację DOS-u na lata.

11. Bariera 640 kB i wojny o pamięć

Żaden rozdział nie oddaje absurdu i pomysłowości epoki DOS-u lepiej niż **wojny o pamięć**. To temat, na który weterani potrafią mówić godzinami — z mieszaniną traumy i nostalgii.

Korzeń problemu tkwił w architekturze procesora 8088 i decyzjach projektowych IBM. Z miliona bajtów adresowalnych w trybie rzeczywistym pierwsze **640 kB** przeznaczono na pamięć programów (tzw. pamięć konwencjonalna), a resztę zarezerwowano na potrzeby systemu (BIOS, pamięć kart graficznych itd.). Tak narodziła się słynna **bariera 640 kB** — granica, o której (apokryficznie) miało paść zdanie „640 kB powinno wystarczyć każdemu”. Bardzo szybko nie wystarczało nikomu.

Wokół tej bariery wyrosła cała barokowa ekonomia pamięci:

- **Pamięć konwencjonalna** — pierwsze 640 kB, w których musiał zmieścić się DOS, sterowniki i sam program. O każdy kilobajt toczono boje.
- **Pamięć górna (UMB)** — wolne fragmenty obszaru między 640 kB a 1 MB, do których dało się „upchnąć” sterowniki i TSR-y, uwalniając cenną pamięć konwencjonalną.
- **Obszar wysoki (HMA)** — pierwsze ~64 kB powyżej 1 MB, dostępne dzięki kwirkowi adresowania (tzw. bramka A20). Od DOS 5.0 dało się tam załadować większość samego jądra (`DOS=HIGH`).
- **Pamięć ekspandowana (EMS)** — sztuczka sprzętowo-programowa (standard LIM EMS), „doklejająca” dodatkową pamięć przez okienko w obszarze górnym.
- **Pamięć rozszerzona (XMS)** — pamięć powyżej 1 MB, zarządzana przez sterownik `HIMEM.SYS`.

Narzędziami tej wojny były sterowniki **HIMEM.SYS** (zarządca pamięci rozszerzonej i bramki A20) oraz **EMM386.EXE** (udostępniający pamięć górną i emulujący EMS na procesorach 386), a także legendarne narzędzia firm trzecich, jak **QEMM** firmy Quarterdeck. Polecenia **LOADHIGH** (**LH**) i **DEVICEHIGH** pozwalały ładować programy i sterowniki do pamięci górnej. Sztuką absolutną było tak ułożyć **CONFIG.SYS**, by zostawić jak najwięcej z 640 kB dla gry czy aplikacji.

```
C:\>MEM /C
```

Modules using memory below 1 MB:

Name	Total	Conventional	Upper Memory
MSDOS	14365 (14K)	14365 (14K)	0 (0K)
HIMEM	1120 (1K)	1120 (1K)	0 (0K)
EMM386	3120 (3K)	3120 (3K)	0 (0K)
MOUSE	17296 (17K)	0 (0K)	17296 (17K)
COMMAND	2624 (3K)	2624 (3K)	0 (0K)

Largest executable program size: 615408 (601K)

Largest free upper memory block: 43792 (43K)

“**☐ Ciekawostka.** Cała ta gimnastyka istniała wyłącznie dlatego, że DOS pozostał w trybie rzeczywistym procesora — tym samym, którego ograniczenia OS/2 i Windows próbowały ominąć trybem chronionym. Gdy w jednym z poprzednich tomów tej serii czytaliśmy o „grzechu pierwotnym 286” w OS/2, mowa była o tej samej barierze. DOS nie ominął jej nigdy — po prostu nauczył się z nią żyć, coraz bardziej pomysłowo i coraz bardziej rozpaczliwie.

12. Wersje 4.0 i 5.0: potknięcie i odkupienie

Po znakomitej linii 3.x przyszło potknięcie. **DOS 4.0** (1988) próbował być nowoczesny — dorzucił graficzną nakładkę **DOS Shell** (menu zamiast gołej linii poleceń) i obsługę partycji większych niż 32 MB — ale zaśląnął głównie jako wersja **ociężała i błędogenna**, zjadająca więcej cennej pamięci konwencjonalnej i sprawiająca problemy ze zgodnością. Wielu użytkowników po prostu została przy sprawdzonym 3.3.

Odkupieniem był **DOS 5.0** (czerwiec 1991) — jedna z najlepszych wersji w dziejach systemu. Jej szandarową cechą było sprytne zarządzanie pamięcią: dzięki `DOS=HIGH` większość jądra ładowała się do obszaru wysokiego (HMA), uwalniając pamięć konwencjonalną dla programów. Doszły też świetny pełnoekranowy edytor **EDIT** (koniec ery prymitywnego, liniowego `EDLIN`), interpreter języka **QBasic** (zastępujący wiekowy GW-BASIC), system pomocy online, narzędzia `UNDELETE` i `UNFORMAT` oraz przełącznik zadań w DOS Shell. DOS 5.0 był dopracowany, oszczędny i lubiany — ale jego premiera miała też drugie dno, o którym za chwilę.

13. Wojna z Digital Research: DR-DOS, FUD i kod AARD

Gary Kildall i Digital Research nie zniknęli po klęsce CP/M-86. W 1988 roku DRI wypuściło **DR-DOS** — system w pełni zgodny z MS-DOS, ale **technicznie go wyprzedzający**. To gorzka ironia całej tej historii: firma, którą DOS pokonał, zaczęła teraz robić *lepszego DOS-a niż sam Microsoft*.

DR-DOS 5.0 (maj 1990) oferował zaawansowane zarządzanie pamięcią i funkcje, które w świecie MS-DOS były dostępne tylko jako płatne dodatki firm trzecich. Microsoft, zaskoczony, zareagował błyskawicznie — i niekoniecznie uczciwie. Zamiast od razu wydać konkurencyjny produkt, **z wyprzedzeniem zapowiadał** nadchodzące, cudowne MS-DOS 5.0, choć do premiery było jeszcze ponad rok. Ta technika — celowe sianie „**FUD**” (*fear, uncertainty, and doubt* — strachu, niepewności i wątpliwości) — skutecznie zamroziła sprzedaż DR-DOS: po co kupować coś teraz, skoro „lada chwila” nadejdzie lepszy MS-DOS? Powtórzono to przy DR-DOS 6.0 i preanonsie MS-DOS 6.0.

Najgłośniejszy był jednak **kod AARD**. W wersjach beta Windows 3.1 Microsoft umieścił zaciemniony, celowo ukryty fragment kodu, który wykrywał, czy Windows działa na **nie-microsoftowym** DOS-ie (np. DR-DOS), i w takim wypadku wyświetlał **niepokojący, choć nieszkodliwy komunikat o błędzie**. Technicznie nic się nie psuło — chodziło wyłącznie o zasianie wątpliwości, czy DR-DOS „naprawdę” jest zgodny. To był podręcznikowy przykład strategii przypisywanej Microsoftowi: *embrace, extend, extinguish* (przygarnij, rozszerz, unicestwij).

Digital Research zostało przejęte przez Novell (powstał **Novell DOS 7**), a prawa trafiły później do firmy **Caldera**, która pozwała Microsoft o praktyki monopolistyczne wymierzone w DR-DOS. Sprawa zakończyła się w 2000 roku ugodą — Microsoft zapłacił około **150 milionów dolarów**. Dla DR-DOS było to jednak pośmiertne zwycięstwo: system od dawna nie liczył się już na rynku.

“ **Ciekawostka — tragiczny finał Kildalla.** Gary Kildall, człowiek, który stworzył pierwowzór całego tego świata, zmarł w 1994 roku w wieku 52 lat, w okolicznościach związanych z upadkiem w barze. Do końca życia czuł się

ograbiony z należnego mu miejsca w historii. Wieloletnia narracja branży — że „przegrał, bo latał samolotem” — była nie tylko nieprawdziwa, ale i okrutna wobec człowieka, którego idee napędzały komputery osobiste, gdy DOS był jeszcze tylko prowizorką.

14. Wersje 6.x i wojna o kompresję: sprawa Stac

Na początku lat 90. twarde dyski były małe i drogie, więc na rynku zatriumfowały programy do **kompresji dysku w locie** — przezroczyście kompresujące i dekompresujące dane, by „podwoić” pojemność. Liderem był **Stacker** firmy **Stac Electronics**, oparty na opatentowanej technologii.

DOS 6.0 (marzec 1993) był naszpikowany nowościami: kompresja dysku **DoubleSpace**, optymalizator pamięci **MemMaker** (automatyzujący koszmarny z rozdziału 11), antywirus **MSAV**, defragmentator **DEFRAG** i program kopii zapasowych **MSBACKUP**. Problem w tym, że **DoubleSpace** **podjejrzenie przypominał Stackera** — gdy beta DOS 6.0 trafiła do biura Stac, firma rozpoznała własną technologię i pozwała Microsoft o naruszenie patentów. Był to **pierwszy proces patentowy w historii Microsoftu**.

W 1994 roku ława przysięgłych przyznała rację Stacowi. Upokorzony Microsoft musiał wydać **DOS 6.21** — wersję, której jedynym celem było **usunięcie DoubleSpace** z komputerów. Wkrótce potem nadszedł **DOS 6.22** (czerwiec 1994) z napisaną od nowa, nienaruszającą patentów kompresją **DriveSpace**. Microsoft dogadał się ze Stakiem (płacąc odszkodowanie i obejmując udziały w firmie), ale wizerunkowo była to porażka.

DOS 6.22 przeszedł do historii jako **ostatnia samodzielna, sprzedawana detalicznie wersja MS-DOS**. Co znamienne, konkurencyjny **IBM PC DOS 7** (1995) był uważany za technicznie ciekawszy — miał wbudowany interpreter REXX i Stackera zamiast DriveSpace. Ale los DOS-u jako samodzielnego produktu był już przesądzony: nadchodził Windows 95.

15. Windows na grzbiecie DOS-u

Przez całe lata 80. i znaczną część 90. **Windows nie był osobnym systemem — był nakładką uruchamianą na DOS-ie**. Ta relacja jest kluczem do zrozumienia, jak DOS jednocześnie tracił widoczność i pozostawał wszechobecny.

Windows 1.0 (1985) i **2.0** (1987) były komercyjnymi niewypałami — toporne graficzne środowiska, które niewielu traktowało poważnie. Dopiero **Windows 3.0** (1990), a potem **3.1**

(1992), odniosły wielki sukces i zaczęły zmieniać peceta w maszynę graficzną. Ale pod spodem wciąż siedział DOS: to on startował maszynę i ładował Windows, który dopiero przełączał procesor w tryb chroniony. Windows był gościem; DOS — gospodarzem.

Ta zależność trwała zaskakująco długo. Cała konsumencka linia **Windows 9x — Windows 95** (z DOS-em 7.0, dodającym długie nazwy plików), **Windows 98** (DOS 7.1, z FAT32) i **Windows Me** (DOS 8.0) — wciąż startowała przez DOS, by następnie załadować właściwe, graficzne jądro w trybie chronionym. Microsoft starał się to ukryć (w Windows 95 tryb „czystego DOS-u” coraz głębiej chowano), ale fundament pozostawał ten sam. DOS został **zdegradowany do roli programu rozruchowego** i warstwy zgodności dla starych aplikacji — niewidoczny, lecz wciąż obecny pod każdym pecetem domowym.

“ **Ciekawostka.** Gdy w Windows 95 wpisywało się polecenie sprawdzające wersję, system pod maską wciąż raportował aplikacjom DOS-owym numer 7.00. Graficzny, „nowoczesny” Windows 95 stał — dosłownie — na grzbiecie wciąż żywego DOS-u. Dla aplikacji DOS-owych ta warstwa była realnym systemem; reszta była tylko piękną dekoracją na wierzchu.

16. Zmierzch: linia NT pożera DOS-a

Tu domyka się wątek, który łączy wszystkie trzy tomy tej serii. Bo systemem, który ostatecznie zabił DOS-a, była **linia Windows NT** — ta sama, która wchłonęła idee VMS-a (przez Dave'a Cutlera) i fragmenty OS/2 (po rozwodzie z IBM).

Microsoft prowadził przez lata **dwie równoległe linie** systemów. Pierwsza — konsumencka, oparta na DOS-ie (Windows 3.x, 95, 98, Me) — była zgodna ze wszystkim, ale dziedziczyła kruchość i ograniczenia DOS-u. Druga — **Windows NT**, zaprojektowana od zera jako prawdziwy, chroniony, 32-bitowy system — **nie miała pod spodem żadnego DOS-u**. Stary, realnotrybowy fundament zastąpiono solidną architekturą rodem z VMS-a. Aplikacje DOS-owe NT uruchamiał w izolowanej maszynie wirtualnej (**NTVDM**) — emulując DOS-a zamiast na nim stać.

Przez całe lata 90. te dwie linie żyły obok siebie: NT dla serwerów i stacji roboczych, linia DOS/9x dla domu. Ostatnim systemem konsumenckim opartym na DOS-ie był **Windows Me** (2000), zawierający MS-DOS 8.0 — i powszechnie uznawany za jeden z najśłabszych Windowsów w historii, ostatnie tchnienie wysłużonego fundamentu.

Przełom nadszedł w **2001 roku** wraz z **Windows XP**. Microsoft dokonał wreszcie tego, co planował od lat: **zjednoczył obie linie na wspólnym jądrze NT**. Domowi i firmowi użytkownicy trafili na ten sam, pozbawiony DOS-u fundament. To był formalny koniec epoki: DOS przestał być podstawą

czegokolwiek, co Microsoft sprzedawał. System, który zbudował imperium z przypadku, został wyparty przez system zbudowany z premedytacją — i to przez system, który nosił w sobie geny dwóch wielkich przegranych, VMS-a i OS/2.

“ **Klamra serii.** Spójrzmy na to z lotu ptaka. Dave Cutler zaprojektował VMS w DEC, a potem Windows NT w Microsoftcie. Microsoft i IBM wspólnie budowali OS/2, którego „wersja 3.0” stała się Windows NT. A DOS — kupiony za 75 tysięcy dolarów następca CP/M — został przez tego właśnie Windows NT ostatecznie zastąpiony. Trzy systemy, trzy zupełnie różne losy, jeden zwycięzca, który wziął geny od wszystkich.

17. Aplikacje, które trzymały DOS przy życiu

System operacyjny jest tyle wart, ile programy, które na nim działają — i to jest klucz do zrozumienia, dlaczego DOS, mimo technicznego ubóstwa, pozostał nie do ruszenia przez tak długo. To nie DOS sprzedawał komputery. Sprzedawały je **aplikacje**, które działały tylko na nim.

Pierwszym prawdziwym „zabójcą” był arkusz kalkulacyjny **Lotus 1-2-3** (1983). To on bardziej niż cokolwiek innego sprzedał peceta IBM światu biznesu: firmy kupowały komputer, żeby uruchomić 1-2-3, a 1-2-3 chodził na DOS-ie. Podobną rolę odegrały edytor **WordPerfect** (długo dominujący w przetwarzaniu tekstu), baza danych **dBASE** (i cały ekosystem języka xBase), arkusz **SuperCalc** czy później **Quattro Pro**. Do tego dochodziły legendarne zestawy narzędzi systemowych — **Norton Utilities** (z kultowym programem do odzyskiwania skasowanych plików) i **PC Tools** — oraz nakładki ułatwiające życie, jak **Norton Commander** (dwupanelowy menedżer plików, którego układ kopiowano przez dekady, aż po dzisiejsze Total Commander i Midnight Commander).

Ta biblioteka oprogramowania stworzyła **efekt zamknięcia** identyczny z tym, który omawialiśmy przy OS/2 — tyle że tym razem działający na korzyść DOS-u. Firma, która zainwestowała w licencje, szkolenia i dane w formatach 1-2-3 czy dBASE, nie miała żadnego powodu, by zmieniać system. DOS był nudny, ale wszystko, czego potrzebował biznes, na nim działało. To wystarczyło, by przez dekadę trzymać konkurencję na dystans.

18. Złoty wiek gier DOS

Jest jeden obszar, w którym DOS nie był nudnym narzędziem pracy, lecz areną prawdziwej magii: **gry**. Dla całego pokolenia DOS to nie arkusze kalkulacyjne, lecz zapach przygody uruchamianej z linii poleceń.

Paradoksalnie, to właśnie **prymitywizm** DOS-u czynił go świetną platformą do gier. Brak ochrony pamięci i pośredniczących warstw oznaczał, że program miał **bezpośredni, niczym nieograniczony dostęp do sprzętu** — mógł rozmawiać wprost z kartą graficzną i dźwiękową, wyciskając z maszyny każdą kroplę wydajności. Dla twórców gier była to wolność, jakiej nie dawał żaden „porządny”, chroniony system.

Era ta dała kanon, który zna każdy: **Doom** i **Wolfenstein 3D** od id Software, **Commander Keen**, **Duke Nukem**, przygodówki **LucasArts** i **Sierra**, symulatory i strategię. Wokół DOS-owych gier wyrosła też **scena demo** (demoscene) — subkultura programistów rywalizujących o to, kto wycisnie z maszyny najbardziej oszałamiające efekty audiowizualne w najmniejszej liczbie bajtów.

Osobnym rozdziałem był dźwięk. Ponieważ DOS nie miał żadnej warstwy abstrakcji audio, każda gra musiała „znać” konkretną kartę dźwiękową — i tak **Sound Blaster** firmy Creative stał się de facto standardem, a ekran konfiguracji, na którym ręcznie wpisywało się adres portu, przerwanie IRQ i kanał DMA karty dźwiękowej, to wspomnienie wspólne dla całego pokolenia graczy.

“ **Ciekawostka.** Dziedzictwo gier DOS jest dziś podtrzymywane głównie przez emulator **DOSBox**, który symuluje cały pecet z DOS-em (wraz z kartą Sound Blaster) i pozwala uruchamiać klasyki na współczesnym sprzęcie. To właśnie DOSBox napędza większość „dosowych” gier sprzedawanych dziś na platformach takich jak GOG.com. DOS jako narzędzie pracy umarł — ale DOS jako maszyna do gier żyje w najlepszej, opakowany w emulator.

19. Programowanie pod DOS-em

Dla programisty DOS był światem zarazem wyzwajająco prostym i frustrująco ciasnym. Z jednej strony — pełny dostęp do maszyny i minimum przeszkód. Z drugiej — wszechobecne ściany 16-bitowej architektury i 640 kB.

Najbliżej sprzętu pracowało się w **assemblerze x86**, wołając usługi systemu i BIOS-u przez przerwania (**INT 21h** dla DOS-u, **INT 10h** dla ekranu, **INT 13h** dla dysku i tak dalej). Wyżej królowały kompilatory, które stały się legendami: **Turbo Pascal** i **Turbo C** firmy Borland (z ich kultowym, błyskawicznym środowiskiem IDE w niebiesko-białej kolorystyce), **Microsoft C** oraz QuickBASIC. To na DOS-ie całe pokolenie nauczyło się programować.

```
; Klasyczne "Hello" w assemblerze x86 dla DOS-u (INT 21h)
.MODEL SMALL
.STACK 100h
.DATA
msg DB 'Witaj, DOS!', 0Dh, 0Ah, '$'
.CODE
```

```
start: MOV AX, @DATA
      MOV DS, AX
      MOV AH, 09h      ; funkcja 09h = wypisz lancuch
      MOV DX, OFFSET msg
      INT 21h          ; wywolaj uslugi DOS-u
      MOV AH, 4Ch      ; funkcja 4Ch = zakoncz program
      INT 21h
      END start
```

Z czasem ciasnota 16-bitowego trybu rzeczywistego stała się dla poważnych programów nie do zniesienia. Odpowiedzią były **rozszerzacze DOS-u** (*DOS extenders*) — sprytna technologia, która pozwalała programowi DOS-owemu przełączyć procesor w 32-bitowy tryb chroniony, zaadresować całą dostępną pamięć (omijając barierę 640 kB), a mimo to wciąż korzystać z usług DOS-u. Standaryzowały to interfejsy **DPMI** i **VCPI**. To właśnie dzięki rozszerzaczom mogły powstać tak wymagające programy jak **Doom** — formalnie „dosowy”, lecz w istocie 32-bitowy. Wśród narzędzi tej epoki błyszcząły kompilatory **Watcom C/C++** (ulubieniec twórców gier) oraz darmowy, oparty na GCC zestaw **DJGPP**, dający na DOS-ie pełnię 32-bitowego programowania w C.

20. Skrypty wsadowe: programowanie dla każdego

Nie sposób opowiadać o DOS-ie bez ukłonu w stronę **plików wsadowych** (*batch files*) — skromnych skryptów `.BAT`, które dla milionów ludzi były pierwszym w życiu kontaktem z „programowaniem”. Nie wymagały kompilatora ani wiedzy — wystarczył zwykły edytor i kilka poleceń. A jednak pozwalały automatyzować realną pracę: menu startowe, kopie zapasowe, instalatory, konfigurację środowiska.

Język był ubogi, ale wystarczający: zmienne środowiskowe, etykiety i skoki (`GOTO`), warunki (`IF`), pętle po plikach (`FOR`), parametry wywołania (`%1`, `%2`) i sterowanie echem. Sztuką było wycisnąć z tego zaskakująco wiele.

```
@ECHO OFF
REM Proste menu startowe w pliku wsadowym
:MENU
CLS
ECHO =====
ECHO  1. Uruchom edytor
ECHO  2. Uruchom arkusz
ECHO  3. Kopia zapasowa dokumentow
```

```
ECHO 4. Wyjście
ECHO =====
SET /P WYBOR=Twoj wybor:
IF "%WYBOR%"=="1" GOTO EDYTOR
IF "%WYBOR%"=="2" GOTO ARKUSZ
IF "%WYBOR%"=="3" GOTO KOPIA
IF "%WYBOR%"=="4" GOTO KONIEC
GOTO MENU

:KOPIA
XCOPY C:\DOKUMENTY\*. * D:\BACKUP\ /S /Y
ECHO Kopia gotowa!
PAUSE
GOTO MENU

:KONIEC
ECHO Do zobaczenia!
```

Bezpośrednim spadkobiercą tej tradycji jest wiersz poleceń Windows (`cmd.exe`), który do dziś wykonuje pliki `.BAT` niemal identycznie. Gdy współczesny administrator pisze skrypt wsadowy w Windows, posługuje się językiem, którego korzenie sięgają wprost `COMMAND.COM` z 1981 roku.

21. Życie po śmierci: FreeDOS i wieczność

Jak w każdym tomie tej serii — gdy oficjalny producent zamyka rozdział, zaczyna się życie po śmierci. Z DOS-em jest ono wyjątkowe, bo nie podtrzymuje go pojedyncza firma na licencji, lecz **ruch open source**.

29 czerwca 1994 roku, gdy Microsoft ogłosił, że przestaje rozwijać i sprzedawać MS-DOS, student **Jim Hall** opublikował manifest proponujący stworzenie wolnej, publicznej wersji DOS-u — początkowo nazwanej **PD-DOS**, wkrótce przemianowanej na **FreeDOS**. Dołączyli inni programiści (m.in. Pat Villani, autor jądra, i Tim Norman). Z poskładanego kodu powstał kompletny, zgodny z MS-DOS system na licencji GNU GPL. Pierwsze stabilne wydanie, **FreeDOS 1.0**, ukazało się dopiero w 2006 roku — bo, jak żartują twórcy, DOS „przestał być ruchomym celem” już w 1994, więc nie ma pośpiechu.

FreeDOS jest dziś czymś więcej niż muzealnym eksponatem. To w pełni utrzymywany, nowoczesny w swojej niszy system: obsługuje **FAT32** i duże dyski (LBA), ma menedżer pakietów w stylu

linuksowym, zestaw narzędzi sieciowych **mTCP** (TCP/IP dla DOS-u!), emulatorzy karty Sound Blaster do gier na nowym sprzęcie i mnóstwo otwartego oprogramowania. Najnowsze wydanie, **FreeDOS 1.4**, ukazało się w kwietniu 2025 roku — ponad trzy dekady po narodzinach idei.

Do czego służy DOS w XXI wieku? Do trzech rzeczy przede wszystkim: **uruchamiania starego, krytycznego oprogramowania** (w przemyśle wciąż pracują maszyny sterowane DOS-em), **aktualizacji firmware'u i BIOS-u** (wiele takich narzędzi to programy DOS-owe — niejeden zrobił sobie kiedyś „dosowy” pendrive do flashowania BIOS-u) oraz **retro-grania**. Twórcy FreeDOS otwarcie przyznają, że na współczesnym sprzęcie najlepiej uruchamiać go w **maszynie wirtualnej** — bo nowe komputery z firmware'em UEFI i procesorami ARM coraz częściej nie potrafią już wystartować klasycznego DOS-u na „gołym metalu”. Tryb rzeczywisty, fundament całej tej historii, powoli znika ze sprzętu.

“ **Ciekawostka.** Maskotką FreeDOS jest rybka **Blinky**. To drobny, ale wymowny znak: system, który narodził się jako bezduszna prowizorka ratująca sprzedaż płyt głównych, dorobił się na starość własnej, sympatycznej maskotki i oddanej społeczności. DOS przeżył wszystkich swoich komercyjnych właścicieli i wciąż dostaje nowe wydania — tyle że teraz robią je ludzie, którzy go po prostu kochają.

22. Epilog: dlaczego DOS wciąż jest wszędzie

DOS jest najlepszym dowodem na to, że w technologii historia rzadko bywa sprawiedliwa. Nie był najlepszym systemem swojej epoki — był jednym z najsłabszych. Nie był oryginalny — był kopią CP/M, napisaną „na szybko i byle jak”. A jednak wygrał, i to nie na chwilę, lecz na całą epokę.

Wygrał, bo znalazł się we właściwym miejscu (pecet IBM), we właściwym czasie (eksplozja rynku), z właściwą ceną (40 dolarów wobec 240 za CP/M) i — przede wszystkim — z właściwą umową (niewyłączna licencja, która oddała Microsoftowi cały rynek klonów). To zwycięstwo dystrybucji i kontraktu nad inżynierią. Gdy zestawimy go z dwoma poprzednimi tomami tej serii, układa się z tego spójna, niepokojąca lekcja: **VMS i OS/2 udowodniły, że doskonałość nie wystarcza, a DOS udowodnił, że często wcale nie jest potrzebna.**

A jednak DOS jest też wszędzie wokół nas — nie jako działający system, lecz jako **nawyk myślowy**. Litery dysków. Ukośniki wsteczne. Rozszerzenia plików. Pliki konfiguracyjne. Wiersz poleceń, który w Windows wciąż wykonuje pliki `.BAT` składnią z 1981 roku. Cały sposób, w jaki świat wyobraża sobie „rozmowę z komputerem” przez tekst, został ukształtowany przez ten skromny, ułomny system. DOS przegrał z linią NT jako technologia — ale wygrał jako **język**, którym do dziś mówimy o komputerach.

„DOS nie był dobry. Był pierwszy, był tani i był wszędzie. W informatyce to zwykle wystarcza, żeby wygrać.”

23. Ciekawostki zebrane

- **Quick and Dirty dosłownie.** Pierwotna nazwa systemu, QDOS, znaczyła „szybki i byle jaki system operacyjny”. To nie był żart — to był opis metody.
- **Imperium za 75 tysięcy dolarów.** Tyle łącznie zapłacił Microsoft za system, który dał mu monopol wart setki miliardów.
- **Klauzula, która zmieniła świat.** Niewyłączne prawo do licencjonowania DOS-u innym producentom — pozornie drobiazg w 1980 roku — okazało się najważniejszą decyzją biznesową w dziejach oprogramowania.
- **Kopia bije oryginał.** IBM wycenił oryginalny CP/M-86 na 240 dolarów, a kopującego go DOS-a na 40. Sześciokrotna różnica ceny pogrzebała pierwowzór.
- **Legenda o samolocie.** Popularna opowieść, że Gary Kildall „przegapił” IBM, bo latał, jest w dużej mierze mitem — deal rozbił się o spór o tantiemy i NDA, nie o jeden nieobecny dzień.
- **640 kB dla każdego.** Słynne zdanie o tym, że „640 kB wystarczy każdemu”, jest prawdopodobnie apokryfem — ale bariera, której dotyczy, była aż nazbyt realna.
- **Xenix, niedoszły następca.** Microsoft miał własnego Uniksa (Xenix) i planował, że to on, a nie DOS, będzie systemem przyszłości. Oddał go SCO w 1987 — i postawił na DOS oraz Windows.
- **Kod AARD.** Ukryty fragment w becie Windows 3.1 straszył użytkowników DR-DOS fałszywymi błędami — podręcznikowy przykład nieczystej gry rynkowej.
- **Pierwszy proces patentowy Microsoftu.** Sprawę o kompresję DoubleSpace wytoczył gigantowi mały Stac Electronics — i wygrał.
- **Windows 95 = DOS 7.** „Nowoczesny” Windows 95 wciąż startował przez DOS-a i raportował aplikacjom numer wersji 7.00.
- **Geny przegranych w zwycięzcy.** Windows NT, który ostatecznie zabił DOS-a, wziął idee od VMS-a (Cutler) i od OS/2 (rozwód z IBM). DOS przegrał z hybrydą dwóch innych przegranych.
- **Rybka, która przeżyła imperium.** FreeDOS, z maskotką-rybką Blinky, wciąż dostaje wydania (1.4 w 2025) — dekady po tym, jak Microsoft pogrzebał MS-DOS.

24. Oś czasu

Rok	Wydarzenie
~1974	Gary Kildall tworzy CP/M; powstaje Digital Research

Rok	Wydarzenie
1980	Tim Paterson pisze QDOS w Seattle Computer Products; rusza Projekt Chess w IBM; IBM i Digital Research nie dogadują się ws. CP/M-86
1980 (XII)	Microsoft kupuje niewyłączną licencję na 86-DOS za 25 000 USD
1981 (V)	Microsoft zatrudnia Tima Patersona
1981 (27 VII)	Microsoft odkupuje pełne prawa do 86-DOS za 50 000 USD
1981 (12 VIII)	Premiera IBM PC 5150 z PC DOS 1.0
1982	MS-DOS 1.25 — pierwsza wersja licencjonowana poza IBM; pierwsze klony
1983	MS-DOS 2.0 (podkatalogi, dyski twarde, sterowniki) z IBM PC XT; Lotus 1-2-3
1984	MS-DOS 3.0 z IBM PC AT (FAT16)
1985	MS-DOS 3.1 (sieci); Windows 1.0
1987	MS-DOS 3.3 z PS/2; Microsoft oddaje Xenix firmie SCO
1988	MS-DOS 4.0 (ociężały); Digital Research wypuszcza DR-DOS
1990	DR-DOS 5.0; Windows 3.0 odnosi sukces
1991	MS-DOS 5.0 (HMA, EDIT, QBasic)
1993	MS-DOS 6.0 (DoubleSpace, MemMaker); pozew Stac
1994	MS-DOS 6.22 (DriveSpace) — ostatnia samodzielna wersja; Jim Hall ogłasza PD-DOS/FreeDOS; śmierć Gary'ego Kildalla
1995	Windows 95 (na DOS-ie 7.0)
1998	Windows 98 (DOS 7.1, FAT32)
2000	Windows Me (DOS 8.0) — ostatni system konsumencki na DOS-ie; ugoda Caldera–Microsoft (~150 mln USD)
2001	Windows XP — zjednoczenie linii na jądrze NT; formalny koniec ery DOS-u
2006	FreeDOS 1.0
2025	FreeDOS 1.4 — DOS wciąż żyje w społeczności open source

25. Źródła i dalsze lektury

Geneza, CP/M, 86-DOS i zakup przez Microsoft

- 86-DOS — Wikipedia: <https://en.wikipedia.org/wiki/86-DOS>
- Seattle Computer Products — Wikipedia: https://en.wikipedia.org/wiki/Seattle_Computer_Products
- „Microsoft MS-DOS early source code” — Computer History Museum: <https://computerhistory.org/blog/microsoft-ms-dos-early-source-code/>
- „Did Bill Gates Steal the Heart of DOS?” — IEEE Spectrum: <https://spectrum.ieee.org/did-bill-gates-steal-the-heart-of-dos>
- „Quick and Dirty: The story of 86-DOS & MS-DOS” — Lunduke: <https://lunduke.substack.com/p/quick-and-dirty-the-story-of-86-dos>

IBM PC, Projekt Chess i klony

- IBM Personal Computer — Wikipedia: https://en.wikipedia.org/wiki/IBM_Personal_Computer
- IBM PC compatible — Wikipedia: https://en.wikipedia.org/wiki/IBM_PC_compatible
- „The IBM PC: On 12 August 1981, the world changed” — The Register: https://www.theregister.com/2011/08/12/ibm_pc_30_anniversary/

Historia wersji, DR-DOS, AARD i sprawa Stac

- MS-DOS — Wikipedia: <https://en.wikipedia.org/wiki/MS-DOS>
- MS-DOS Version History — DOS Wiki (Fandom): https://dos.fandom.com/wiki/MS-DOS_Version_History
- „Disk Compressor History: A Microsoft Antitrust Prelude” — Tedium: <https://tedium.co/2018/09/04/disk-compression-stacker-doublespace-history/>
- Hasło o DR-DOS i kodzie AARD — Wikipedia: https://en.wikipedia.org/wiki/AARD_code

Windows na DOS-ie, linia NT i FreeDOS

- FreeDOS — Wikipedia: <https://en.wikipedia.org/wiki/FreeDOS>
- The FreeDOS Project (strona oficjalna): <https://www.freedos.org/>
- „FreeDOS 1.4: Still DOS, still FOSS, more modern than ever” — The Register: https://www.theregister.com/2025/04/09/freedos_14/

Uwaga: niektóre daty wydań MS-DOS różnią się między źródłami o kilka miesięcy (np. MS-DOS 6.0 jako marzec lub sierpień 1993), ponieważ inaczej liczono premierę detaliczną, OEM-ową i wydanie dla IBM. Podane kwoty ugód (Caldera, Stac) i ceny pochodzą z materiałów epoki i bywają zaokrąglane.

Revision #1

Created 21 June 2026 19:38:54 by Tomasz Konieczny

Updated 21 June 2026 19:39:01 by Tomasz Konieczny