Tabela systemów i porównanie poleceń

Materiał pomocniczy serii.

Dwa zestawienia obejmujące całą serię, utrzymywane w jednym miejscu i dołączane do poszczególnych tekstów mechanizmem dołączania treści BookStacka. Zmiana wprowadzona tutaj pojawia się we wszystkich tekstach, które te tabele dołączają.

> Ta strona jest materiałem pomocniczym serii, a nie osobną opowieścią. Jeżeli trafiłeś tu z odsyłacza w środku artykułu — to znaczy, że czytasz właśnie źródło tabeli, która była tam wyświetlona.

□□ Skróty użyte na tej stronie (49)

Skrót	Co znaczy
AARD	inicjały programisty Microsoftu; kod w Windows 3.1 wyświetlający niejasne ostrzeżenie pod DR DOS-em
AI	Artificial Intelligence: sztuczna inteligencja; akcelerator AI ma procesor maszyny z17
AROS	AROS Research Operating System: otwarty system zgodny z AmigaOS
BIOS	Basic Input/Output System: warstwa oddzielająca system od sprzętu, pomysł z CP/M
BSD	Berkeley Software Distribution: odmiana Uniksa z Uniwersytetu Kalifornijskiego w Berkeley
CIO	Central Input/Output: nazwana abstrakcja urządzeń w systemie ośmiobitowych Atari
CLIST	Command List: starszy język skryptowy TSO
CMD	Command: interpreter poleceń OS/2 i Windows
CP/M	Control Program for Microcomputers: system Gary'ego Kildalla dla mikrokomputerów
DCL	Digital Command Language: powłoka i język skryptowy OpenVMS
DD	Data Definition: karta JCL opisująca zbiór używany przez krok zadania
DFDSS	Data Facility Data Set Services: narzędzie z/OS do kopii zapasowych zbiorów i woluminów
DFSORT	Data Facility Sort: program sortujący IBM uruchamiany w JCL-u
DOS	Disk Operating System: tu MS-DOS i jego odmiany

Skrót	Co znaczy
EOD	End of Day: w poleceniu Z EOD — zatrzymanie systemu na koniec dnia
EPM	Enhanced Editor for Presentation Manager: edytor tekstu OS/2
EVE	Extensible Versatile Editor: pełnoekranowy edytor OpenVMS oparty na TPU
HAL	Hardware Abstraction Layer: warstwa ukrywająca przed jądrem Windows NT różnice sprzętu
HLQ	High-Level Qualifier: pierwszy człon nazwy zbioru na z/OS
HPFS	High Performance File System: system plików OS/2
ICKDSF	Device Support Facilities: narzędzie z/OS do inicjalizacji i obsługi dysków
IPL	Initial Program Load: rozruch systemu z/OS; D IPLINFO pokazuje jego parametry
ISPF	Interactive System Productivity Facility: panelowa nakładka na TSO
JCL	Job Control Language: język opisu zadań wsadowych na z/OS
JES	Job Entry Subsystem: podsystem z/OS przyjmujący i kolejkujący zadania wsadowe
MIDI	Musical Instrument Digital Interface: złącze instrumentów muzycznych, standard w Atari ST
NCP	NetWare Core Protocol: protokół plikowy NetWare, przeniesiony na Linuksa
NSS	Novell Storage Services: system plików NetWare, przeniesiony na Linuksa
NT	New Technology: linia Windows NT
OES	Open Enterprise Server: produkt OpenText, w którym żyją usługi NetWare
OS/2	Operating System/2: system IBM i Microsoftu dla pecetów, dziś ArcaOS
PDS	Partitioned Data Set: zbiór-biblioteka z nazwanymi członkami na z/OS
PID	Process Identifier: identyfikator procesu
POSIX	Portable Operating System Interface: standard interfejsu systemowego wzorowany na Uniksie
RACF	Resource Access Control Facility: menedżer bezpieczeństwa z/OS
REXX	Restructured Extended Executor: język skryptowy IBM, obecny na z/OS i OS/2
RMS	Record Management Services: warstwa rekordowego dostępu do plików OpenVMS
ROM	Read-Only Memory: pamięć stała, w której siedział system Atari
SCO	The Santa Cruz Operation: firma uniksowa, której groźba procesów wystarczyła na trzy lata
SDSF	System Display and Search Facility: narzędzie z/OS do kolejek zadań i wydruków
TCP/IP	Transmission Control Protocol / Internet Protocol: protokoły Internetu, dorobek BSD
TOS	The Operating System: system Atari ST
TPU	Text Processing Utility: silnik edytorów OpenVMS
TSO	Time Sharing Option: środowisko pracy interaktywnej na z/OS
TWS	Tivoli Workload Scheduler: harmonogram zadań wsadowych IBM

Skrót	Co znaczy
UEFI	Unified Extensible Firmware Interface: współczesny następca BIOS-u
UIC	User Identification Code: tożsamość właściciela pliku w OpenVMS
VMS	Virtual Memory System: dawna nazwa OpenVMS; VMS Software rozwija go od 2014
z/OS	główny system operacyjny mainframe'ów IBM

Szesnaście systemów, jedna lekcja

Zestawienie obejmuje wszystkie systemy opisane w serii. Kolumna „Co go pogrążyło” i kolumna „Co go uratowało” są celowo rozdzielone, bo w większości przypadków to są zupełnie różne mechanizmy — i to jest właśnie obserwacja, po którą warto tu wracać.

System	Rok	Największa przewaga	Co go pogrążyło	Co go uratowało	Stan w 2026
Mainframe (z/OS)	1964	Kanały wejścia-wyjścia i niezłamana obietnica zgodności przez 60 lat	Nic. Cztery fale przepowiedni o śmierci nie sprawdziły się	Ekonomia niemożliwej migracji	**Żywy i rozwijany.** z17, procesor z akceleratorem AI
UNIX	1969	Kilkanaście prostych pojęć, z których składa się wszystko inne; język C i przenośność	Fragmentacja i pięć lat wojny domowej	Uczelnie, którym rozdano kod za 150 dolarów	Trzy komercyjne warianty w trybie podtrzymania. **Idea wygrała całkowicie**
CP/M	1974	**Rozdzielenie systemu od sprzętu warstwą BIOS** — pomysł, który przeżył go o pół wieku	Jedna nieodebrana szansa w 1980 roku i cena 240 dolarów wobec 40	Nic. Digital Research zniknęło po drodze	Martwy jako produkt. **Żywy w każdej literze dysku i w każdej nazwie 8.3**
OpenVMS	1977	Klastry ze współdzielonym zapisem, dwadzieścia lat przed resztą świata	Kolejni właściciele bez interesu w produkcie	VMS Software (2014)	Żywy niszowo, port na x86-64, wsparcie do 2035

System	Rok	Największa przewaga	Co go pogrążyło	Co go uratowało	Stan w 2026
BSD	1978	Gniazda, TCP/IP, Fast File System — czyli połowa dzisiejszego UNIX-a	Dwa lata niepewności prawnej (1992–1994)	Ugoda sądowa i wydanie 4.4BSD-Lite	**Cztery żywe projekty.** Kod w miliardzie telefonów i w konsolach
Atari (OS ROM i TOS)	1979	Osobny procesor graficzny z własną listą wyświetlania i **nazwana abstrakcja urządzeń** (CIO) — w komputerze domowym	Krach 1983, podział firmy i właściciel, który ciął koszty zamiast inwestować	MIDI w standardzie — Atari ST przeżyło w studiach nagraniowych jeszcze piętnaście lat	Martwy komercyjnie. **Żywy w EmuTOS, FreeMiNT i na scenie 8-bitowej**
DOS	1981	**Żadna techniczna.** Cena, moment i klauzula w umowie	Własny następca z linii NT	Ruch wolnego oprogramowania (FreeDOS)	Żywy w niszy: aktualizacje firmware'u i retro-granie
Commodore (KERNAL)	1982	Własna fabryka krzemu, czyli cena, której nikt nie mógł dogonić — i **tablica skoków KERNAL** jako obietnica zgodności	Zarząd bez pomysłu po odejściu Tramiela i pięć lat bez inwestycji w układy	Nic. Firma zbankrutowała 29 kwietnia 1994	Martwy jako firma. **C64 ma w 2026 czynną scenę i produkowany nowy sprzęt**
Novell NetWare	1983	**Wyspecjalizowany aż do przesady serwer plików** — wszystko w pierścieniu zerowym, cała wolna pamięć jako cache	Ta sama specjalizacja. Gdy „serwer” przestał znaczyć „serwer plików”, NetWare przestał być kategorią	Przeniesienie eDirectory, NCP i NSS na Linuksa	Jądro martwe od 2011. **Katalog i usługi żyją w OpenText OES**
Mac OS / macOS	1984	Interfejs, za który ludzie płacili więcej — i trzy udane zmiany architektury procesora	**Własna architektura.** Brak ochrony pamięci przez 17 lat i trzy skasowane projekty następcy	Kupienie systemu od własnego wygnanego założyciela (NeXT, 1996)	**Żywy i dominujący w swojej niszy.** Jeden z trzech aktywnie certyfikowanych UNIX-ów 03

System	Rok	Największa przewaga	Co go pogrążyło	Co go uratowało	Stan w 2026
AmigaOS	1985	**Wywłaszczająca wielozadaniowość z priorytetami w 1985 roku**, na maszynie bez jednostki zarządzania pamięcią	Brak ochrony pamięci, pięć lat bez nowego chipsetu i właściciel, który nie rozumiał, co ma	Społeczność — system dostaje wydania trzydzieści lat po bankructwie producenta	**Żywy niszowo:** AmigaOS 4, MorphOS, AROS; emulacja i nowy sprzęt
Windows 1.0–3.11	1985	**Działał na sprzęcie, który ludzie już mieli** — i dawał jeden sterownik drukarki dla wszystkich programów	Własny następca. Do końca nie miał ochrony pamięci ani wywłaszczania	Nie potrzebował ratunku — wygrał i został zastąpiony przez swojego producenta	Martwy. Wsparcie dla systemów wbudowanych skończyło się 1 listopada 2008
OS/2	1987	32 bity, obiektowy pulpit i zgodność z DOS-em oraz Windows naraz	Model dystrybucji konkurenta	Arca Noae (2017)	Żywy niszowo. ArcaOS startuje z UEFI
Linux	1991	**Brak wątpliwości prawnych i otwarcie na każdego**	Nic	Nie wymagał ratunku	**Wygrał.** 100% superkomputerów, większość serwerów, Android
Windows NT	1993	**Architektura zaprojektowana od zera przez człowieka, który zrobił to już raz** — podsystemy, HAL, prawdziwy model bezpieczeństwa	Nic. Jedyne poważne potknięcie to Vista	Nie wymagał ratunku	**Żywy i dominujący na pulpicie.** Windows 11, Azure, Xbox
Windows 95–98–Me	1995	Trzydziestodwubitowe programy i wywłaszczanie na maszynie za tysiąc dolarów, bez łamania dekady cudzego oprogramowania	Zgodność wsteczna wpisana w architekturę: kod szesnastobitowy w krytycznych miejscach, sterty 64 KB, zero ochrony pamięci jądra	Nic. Zjednoczenie z linią NT w Windows XP zamknęło ją całkowicie	Martwy od 2006. **Żywy w każdej długiej nazwie pliku i w pasku zadań**

Pięć wniosków

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.

To samo zadanie w pięciu światach

Zestawienie scalone z czterech wcześniejszych wersji tej tabeli — z tekstów o DOS-ie, OS/2, OpenVMS-ie i mainframie. Kolumny stoją w stałej kolejności: od świata uniksowego, przez dwa systemy pecetowe, po dwa systemy z maszyn większych. Trzydzieści osiem czynności, wykonywanych na co dzień, pokazanych obok siebie.

> Zestawienie jest poglądowe. W każdej kolumnie istnieją warianty, przełączniki i narzędzia dodatkowe, których tu nie ma; chodzi o pokazanie, jak te same pojęcia nazywa się w pięciu różnych tradycjach, a nie o kompletną dokumentację żadnej z nich.

Zadanie	Unix / Linux (bash)	DOS / Windows cmd	OS/2 CMD	OpenVMS (DCL)	z/OS (TSO / ISPF / JCL)
Lista plików	`ls -l`	`DIR`	`DIR`	`DIRECTORY`	`LISTCAT` / ISPF 3.4
Lista rekurencyjna	`ls -R`	`DIR /S`	`DIR /S`	`DIR [...]`	`LISTCAT LEVEL(HLQ)`
Drzewo katalogów	`tree`	`TREE`	`TREE`	`DIR [...]/BRIEF`	brak — zbiory są płaskie
Zawartość biblioteki	`ls katalog/`	`DIR katalog`	`DIR katalog`	`DIR [.KATALOG]`	ISPF 3.4 → `M` (członkowie PDS)
Podgląd pliku	`cat` / `less`	`TYPE` / `MORE`	`TYPE`	`TYPE/PAGE`	ISPF Browse
Podgląd logu	`tail -f /var/log/…`	Event Viewer	`TYPE` pliku	`TYPE OPERATOR.LOG`	SDSF `LOG` / SYSLOG
Nowy katalog	`mkdir`	`MD`	`MD`	`CREATE/DIRECTORY [.PROJEKT]`	PDS: alokacja zbioru
Nowy plik / zbiór	`touch`	`COPY CON`	`COPY CON`	`CREATE`	`ALLOC` / `DD DISP=(NEW,CATLG)` *z deklaracją miejsca!*
Kopiowanie	`cp`	`COPY` / `XCOPY`	`COPY` / `XCOPY`	`COPY`	`IEBGENER` (JCL)
Zmiana nazwy	`mv`	`REN`	`REN`	`RENAME`	`ALTER NEWNAME`
Kasowanie	`rm`	`DEL`	`DEL` / `ERASE`	`DELETE plik.txt;3` *(numer wersji obowiązkowy)*	`DELETE` / IDCAMS
Kasowanie drzewa	`rm -rf`	`DELTREE` *(od 6.0)*	brak wbudowanego — narzędzia zewnętrzne	`DELETE [...]*.*;*`	IDCAMS

Zadanie	Unix / Linux (bash)	DOS / Windows cmd	OS/2 CMD	OpenVMS (DCL)	z/OS (TSO / ISPF / JCL)
Zmiana katalogu	`cd projekt`	`CD`	`CD`	`SET DEFAULT [.PROJEKT]`	brak — zbiory są płaskie
Szukanie tekstu	`grep ERROR *.log`	`FIND` / `FINDSTR`	`FIND`	`SEARCH *.LOG "ERROR"`	ISPF `SRCHFOR`
Porównanie plików	`diff`	`FC`	`COMP`	`DIFFERENCES`	ISPF `SUPERC`
Sortowanie	`sort`	`SORT`	`SORT`	`SORT`	`DFSORT` / `SYNCSORT` w JCL-u
Edycja	`vi` / `nano`	`EDIT`	`E` / EPM	`EDIT/TPU` *(EVE)*	ISPF Edit
Uruchomienie programu	`./program`	`program.exe`	`program.exe`	`RUN PROGRAM`	`CALL 'BIBL.LOAD(PROGRAM)'` lub `EXEC PGM=`
Skrypt	`.sh`	`.BAT`	`.CMD` (batch lub REXX)	`.COM` (DCL)	`.CLIST` / REXX / JCL
Zadanie wsadowe	`at` / `batch` / kolejka	Task Scheduler	`START`	`SUBMIT`	`SUB` (wysłanie JCL do JES)
Kolejka zadań	`jobs` / `atq`	`SCHTASKS`	—	`SHOW QUEUE/BATCH`	SDSF: `ST` (status), `H` (held), `O` (output)
Zadanie w tle	`./praca.sh &`	`START /B`	`START /B`	`SUBMIT praca.com` *(kolejka wsadowa)*	`SUB` (JES)
Zadanie cykliczne	`cron`	Task Scheduler	obiekt na pulpicie / `AT`	`SUBMIT/AFTER=` + `$SETIMR`	harmonogram (np. TWS)
Lista procesów	`ps` / `top`	`TASKLIST`	`PSTAT`	`SHOW SYSTEM`	SDSF `DA`
Zabicie procesu	`kill -9 PID`	`TASKKILL`	`KILL`	`STOP/IDENTIFICATION=20A0031C`	SDSF `C` (cancel)
Przekierowanie wyjścia	`> plik.txt`	`> plik.txt`	`> plik.txt`	`/OUTPUT=plik.txt` *(kwalifikator, nie operator)*	`DD SYSOUT=*`
Potok	`cmd1 | cmd2`	`|`	`|`	`PIPE cmd1 | cmd2` *(dołożone dopiero w latach 90.)*	brak w tym sensie
Zmienna środowiskowa	`export X=1`	`SET X=1`	`SET X=1`	`X = "1"` *(symbol)* `DEFINE DANE DKA0:[BAZA]` *(nazwa logiczna)*	`ALLOC` / DD w JCL

Zadanie	Unix / Linux (bash)	DOS / Windows cmd	OS/2 CMD	OpenVMS (DCL)	z/OS (TSO / ISPF / JCL)
Uprawnienia pliku	`chmod 750`	`ATTRIB`	`ATTRIB`	`SET SECURITY/PROTECTION=(S:RWED,O:RWED,G:RE,W)`	RACF `PERMIT`
Właściciel pliku	`chown`	—	—	`SET FILE/OWNER_UIC=`	RACF profile
Konto użytkownika	`useradd`	`NET USER`	`NET USER`	`AUTHORIZE ADD JAN`	RACF `ADDUSER JAN`
Kto jest zalogowany	`who` / `w`	`QUSER`	`NET WHO`	`SHOW USERS`	SDSF `DA`
Formatowanie nośnika	`mkfs.ext4`	`FORMAT`	`FORMAT /FS:HPFS`	`INITIALIZE DKA100: DANE`	`ICKDSF`
Sprawdzenie nośnika	`fsck`	`CHKDSK /F`	`CHKDSK /F`	`ANALYZE/DISK_STRUCTURE/REPAIR`	`IDCAMS EXAMINE`
Kopia zapasowa	`tar` / `rsync`	`XCOPY` / produkty	`BACKUP`	`BACKUP` *(w systemie, świadomy RMS)*	`DFDSS` / `IEBCOPY`
Wersja systemu	`uname -a`	`VER`	`VER`	`SHOW SYSTEM` / `F$GETSYI("VERSION")`	`D IPLINFO`
Pomoc	`man cp`	`COPY /?`	`HELP COPY`	`HELP COPY /LOG` *(pełna, w systemie)*	`HELP` / ISPF F1
Zamknięcie systemu	`shutdown -h now`	menu Start	`SETBOOT` / menu	`@SYS$SYSTEM:SHUTDOWN`	operator: `Z EOD`

Jak tego używać w innych tekstach

Obie tabele mają własne kotwice i można je dołączyć do dowolnej strony w tej przestrzeni:

- `{{@ID#bkmrk-tabela-systemow}}` — samo zestawienie systemów
- `{{@ID#bkmrk-piec-wnioskow}}` — pięć wniosków pod zestawieniem
- `{{@ID#bkmrk-porownanie-polecen}}` — porównanie poleceń

gdzie `ID` to numer tej strony widoczny w jej adresie po zapisaniu.

Revision #9
Created 2026-09-17 11:56:41 UTC by Tomasz Konieczny
Updated 2026-09-27 18:33:16 UTC by Tomasz Konieczny