Systemy operacyjne

Co to jest system operacyjny?

System operacyjny jest **programem**;

działa jako pośrednik między użytkownikiem komputera a sprzętem komputerowym;
tworzy środowisko, w którym użytkownik może wykonywać programy.

System operacyjny nadzoruje i koordynuje posługiwanie się sprzętem przez programy użytkowe, które pracują na zlecenie użytkowników.

Co to jest system operacyjny?

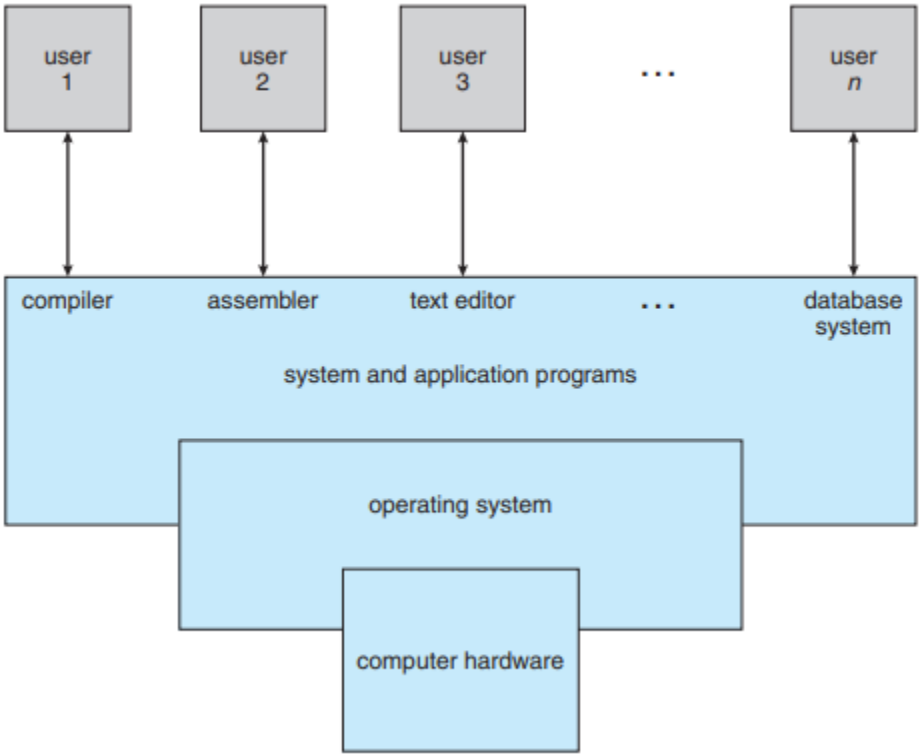
System operacyjny jest odpowiedzialny za

– zarządzanie zasobami komputera

– tworzenie wirtualnej maszyny (dla programisty)

Jądro (kernel) – część systemu operacyjnego, która działa nieustannie; wszystkie pozostałe programy są programami użytkowymi.

Składniki systemu komputerowego



- Sprzęt
- System operacyjny
- Programy

Cechy dobrego systemu operacyjnego

- Funkcjonalność
- wydajność (procesor powinien pracować dla użytkownika)
- skalowalność (zwiększenie obciążenia i rozbudowa sprzętu)
- niezawodność (dostępność, five nines)
- łatwość korzystania i zarządzania

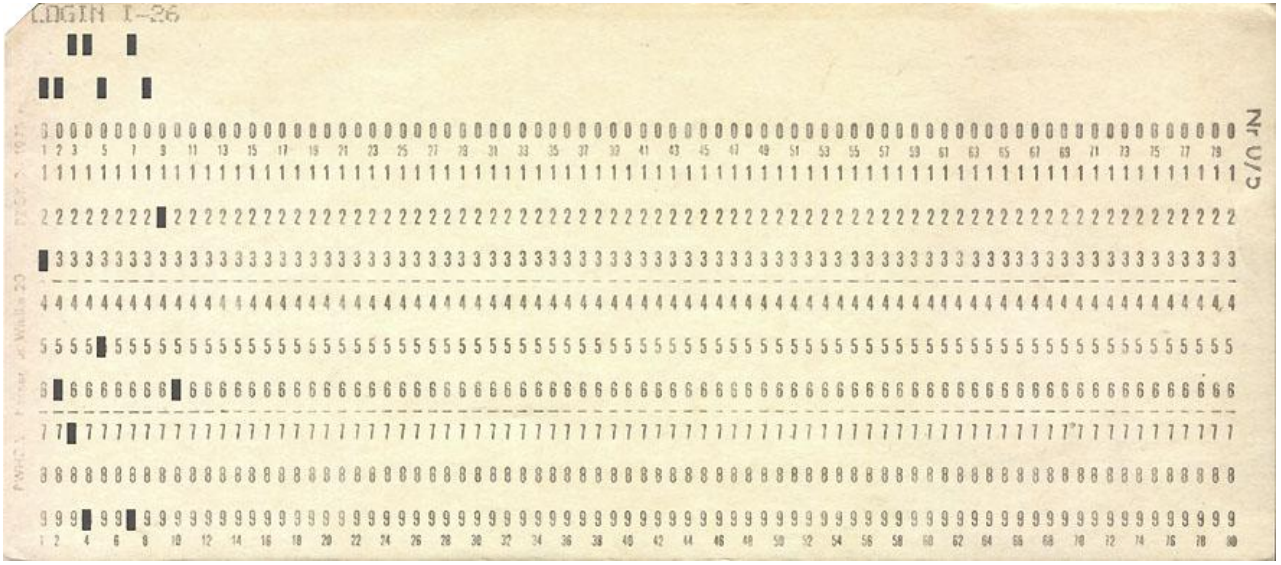
Historia rozwoju komputerów i systemów operacyjnych

- 1642-1945 (0. generacja): mechaniczne maszyny liczące
- 1945-1953 (1. generacja): komputery na lampach próżniowych

– programowanie w języku maszynowym (brak języka programowania)

– przeprowadzanie wyłącznie obliczeń numerycznych

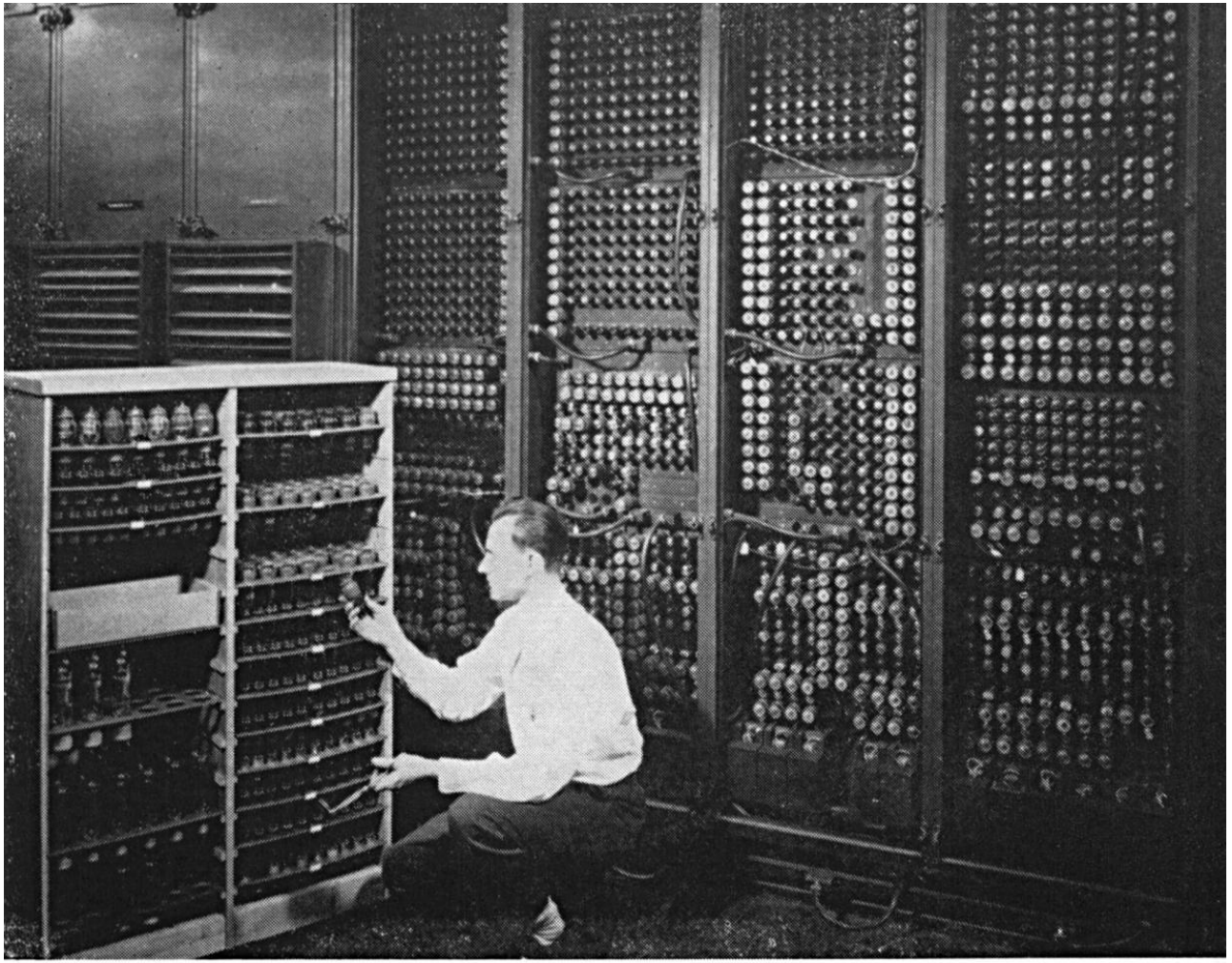
– zastosowanie kart dziurkowanych do wprowadzania danych (początek lat 1950.)



Historia - Eniac

- **ENIAC** Electronic Numerical Integrator and Computer
- Elektroniczny integrator numeryczny i komputer (1946-1955, J.Mauchly, J.P.Eckert):
- 18 tys. lamp próżniowych, 70 tys. oporników, 10 tys. kondensatorów, 1.5 tys. przekaźników, 6 tys. ręcznych przełączników
- 30 ton, 170 m2, moc 160kW
- 5000 operacji dodawania na sekundę
- maszyna dziesiętna (każda liczba była reprezentowana przez pierścień złożony z 10 lamp)
- ręczne programowanie przez ustawianie przełączników i wtykanie kabli

ENIAC



Replacing a bad tube meant checking among ENIAC's 19,000 possibilities.

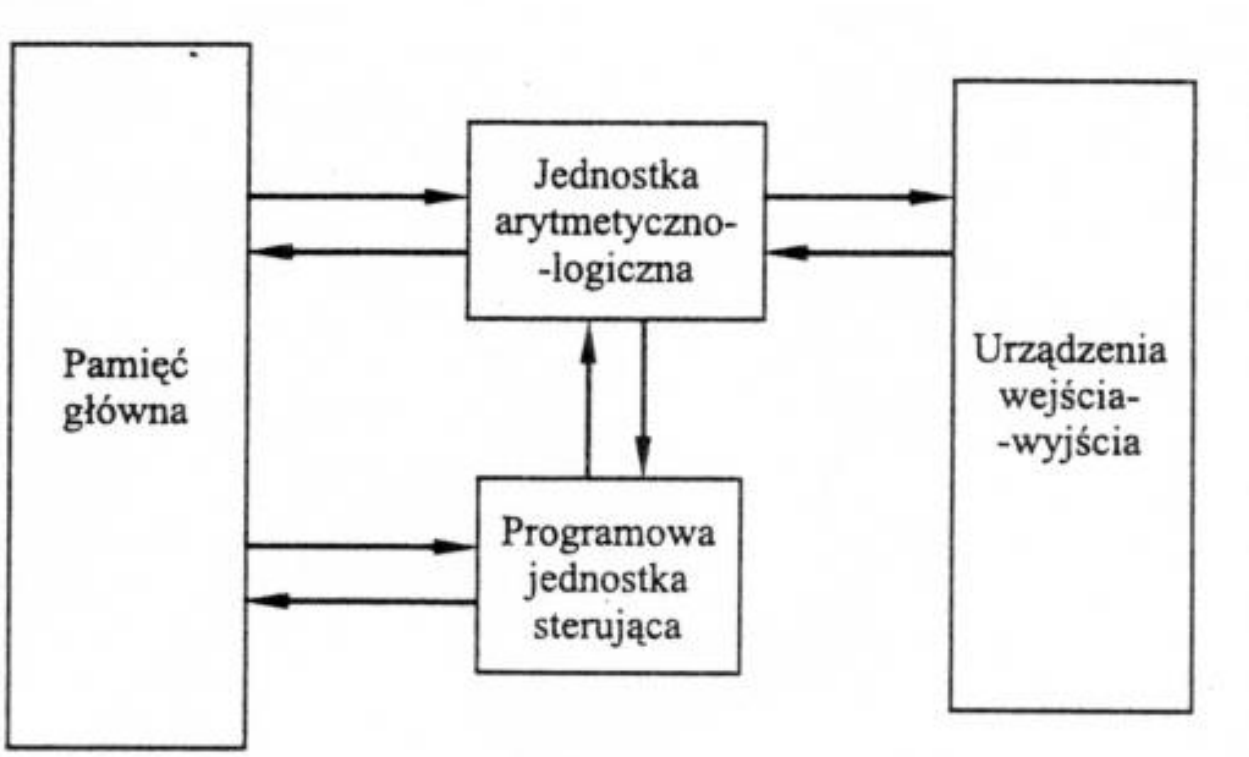
Park w Bletchley

- Park w Bletchley, Anglia, 1939-1945
- Government Code and Cipher School, siedziba wywiadu, gdzie łamano niemieckie szyfry przy pomocy ulepszonej „bomby Rejewskiego”
- (M.Rejewski, J.Różycki, H.Zygalski3
-), a następnie maszyn Heath Robinson, Mark 1 Colossus (1943) i Mark 2 Colossus (1944).

EDVAC Electronic Discrete Variable Computer

- Komputer wg Johna von Neumanna (1946, IAS4) (pomysł Mauchly'ego i Eckerta):
- pamięć główna (przechowywanie danych i rozkazów), 1000 słów 40 bitowych; każdy element pamięci ma unikalny identyfikator (zwykle numer) nazywany adresem
- jednostka arytmetyczno-logiczna (ALU, Arithmetic-Logic Unit) wykonująca działania na danych binarnych
- jednostka sterująca, która interpretuje rozkazy z pamięci i powoduje ich wykonanie
- urządzenia wejścia-wyjścia, których pracą kieruje jednostka centralna

Maszyna von Neumanna



Pierwsze komputery komercyjne

- W roku 1947 Mauchly i Eckert tworzą Eckert-Mauchly Computer Corporation. Powstaje pierwszy komputer komercyjny UNIVAC I(Universal Automatic Computer), który może realizować
  - macierzowe rachunki algebraiczne
  - problemy statystyczne
  - obliczanie premii dla firm ubezpieczeniowych
- UNIVAC I został wykorzystany do powszechnego spisu w 1950 r.
- IBM wyprodukował swój pierwszy komputer elektroniczny z przechowywanym programem do zastosowań naukowych w 1953 r. (model 701). Model 702 (1955) znalazł zastosowanie w biznesie

Komputery tranzystorowe 2 generacja

- 1953-1965 (2. generacja): komputery tranzystorowe

– zwiększona niezawodność

– rozgraniczenie roli operatora i programisty

– system wsadowy (wczytywanie i drukowanie off-line)

– karty sterowania zadaniami (Fortran Monitor System, Job Control Language)

Prosty monitor wraz z językiem kontroli zadań (JCL) umożliwiał przetwarzanie wsadowe (batch processing).
Bardziej złożone jednostki ALU oraz sterujące, ulepszone języki programowania, rozpoczęto dostarczanie oprogramowania systemowego.

Komputery 3 generacja

- 1965-1980 (3. generacja): komputery zbudowane z układów scalonych

– ujednolicenie linii produkcyjnych; małe i duże komputery z ujednoliconym systemem operacyjnym (ta sama lista rozkazów, ale różna wielkość pamięci, szybkość procesorów, wydajność)

– wieloprogramowość, spooling

– systemy z podziałem czasu: Compatible Time-Sharing System (CTSS), Multiplexed Information and Computing Service (MULTICS)

- minikomputery (DEC PDP-8) – 1964 – pierwszy minikomputer

DEC PDP-8

Komputery 4 generacja

- od 1980 (4. generacja): komputery oparte na układach VLSI
- układy o dużym stopniu scalenia (VLSI), powstanie stacji roboczych i komputerów osobistych (architektura minikomputerów)
- sieciowe systemy operacyjne
- rozproszone systemy operacyjne (układy masywnie równoległe)

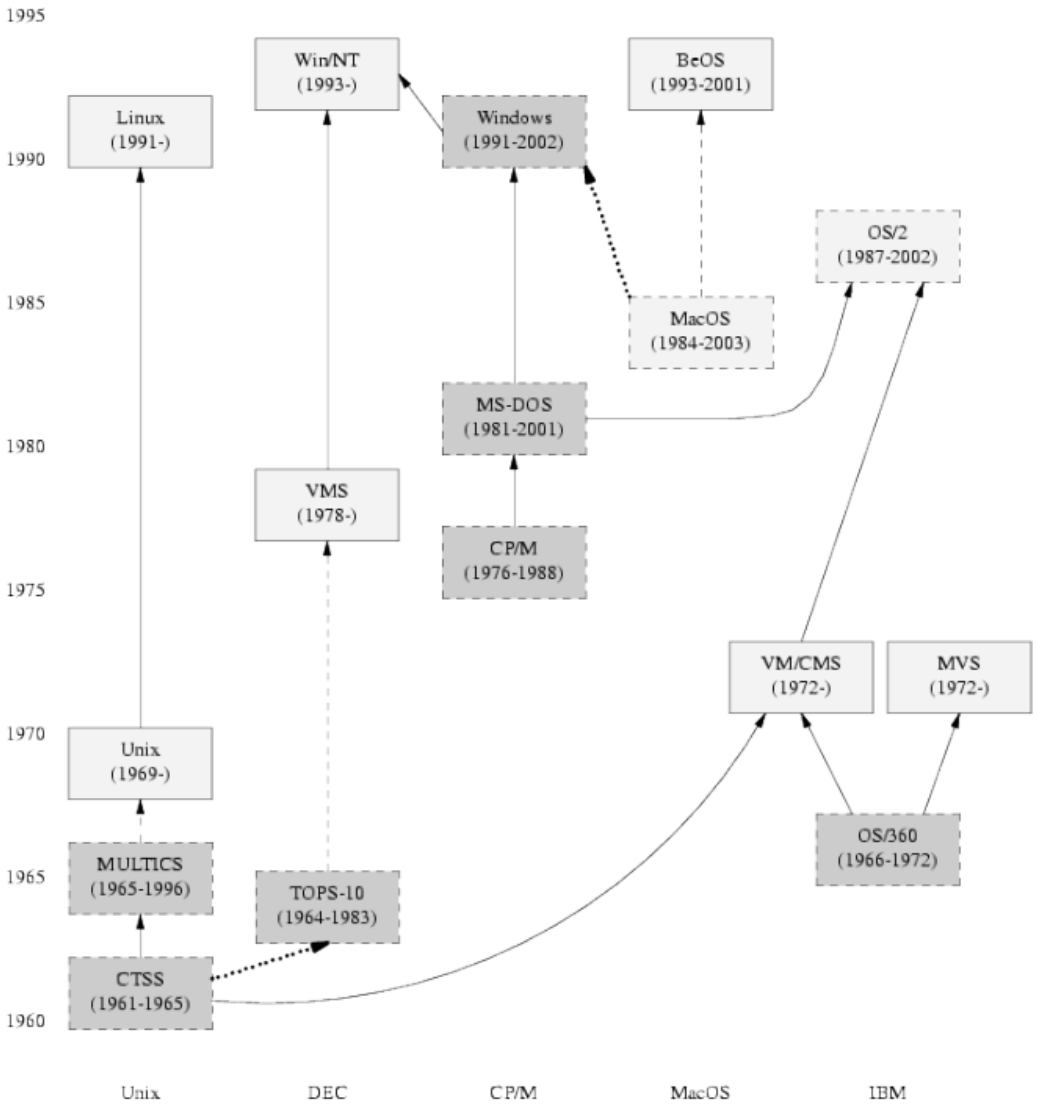
Zalety układów scalonych

- koszt mikroukładu niezmieniony pomimo wzrostu gęstości upakowania
- gęstsze upakowanie to większa szybkość działania układu
- zmniejszenie wymiarów komputera
- mniejsze zapotrzebowanie na moc i łatwiejsze chłodzenie
- połączenia wewnątrz układu scalonego są bardziej niezawodne, niż połączenia lutowane

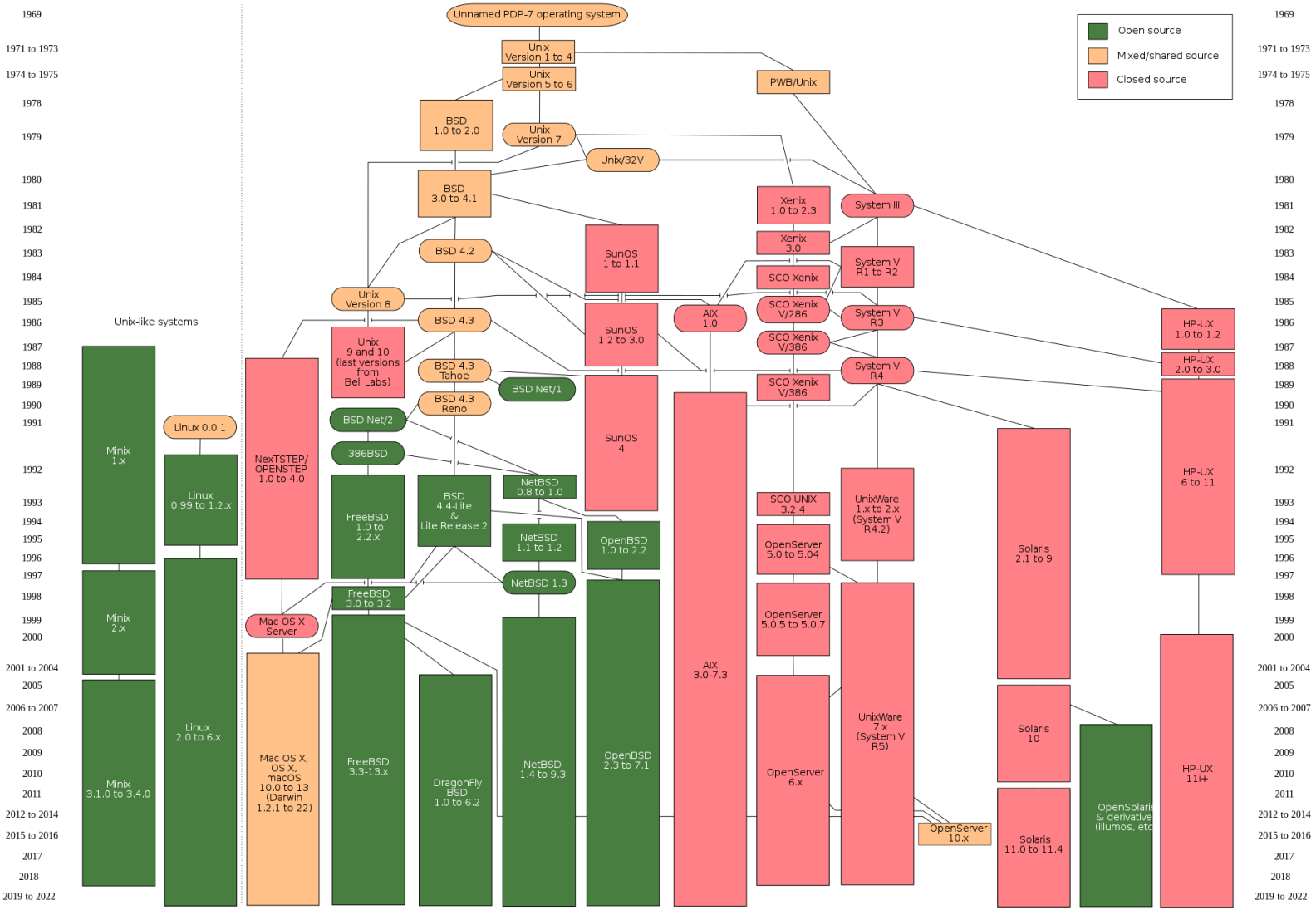
Początki komputeryzacji w Polsce

- w grudniu 1948 roku zaczyna działać w Państwowym Instytucie Matematycznym Grupa Aparatów Matematycznych kierowana przez dra Henryka Greniewskiego
- opracowanie Systemu Automatycznego Kodowania (SAKO, 1960), Polski FORTRAN
- Odra 1002 (1962) – pierwszy komputer z zakładów Elwro, Odra 1003 (1964) – pierwszy seryjnie produkowany komputer (ostatni wyłączono 30/04/2010 po 34 latach pracy)
- 16-bitowy minikomputer K-202 J.Karpińskiego (1970-1973), pierwszy polski komputer zbudowany z użyciem układów scalonych, przewyższał pod względem szybkości pierwsze IBM PC oraz umożliwiał wielozadaniowość, wielodostępność i wieloprocesorowość.
- minikomputer Mera (1978)

Historia systemów operacyjnych wg E.S.Raymonda



Historia systemu UNIX



Ewolucja procesorów firmy Intel

Ewolucja mikroprocesorów firmy Intel†

parametr	8008	8080	8086	80386	80486
rok wprowadzenia	1972	1974	1978	1985	1989
liczba rozkazów	66	111	133	154	235
szerokość szyny adresowej	8	16	20	32	32
szerokość szyny danych	8	8	16	32	32
liczba rejestrów	8	8	16	8	8
adresowalność pamięci	16KB	64KB	1MB	4GB	4GB
szerokość pasma magistrali (MB/s)	-	0.75	5	32	32
czas dodawania rejestr-rejestr (μs)	-	1.3	0.3	0.125	0.06

Mikroprocesory firmy Intel

parametr	286	386	486	Pentium	P6
początek projektowania	1978	1982	1986	1989	1990
rok wprowadzenia	1982	1985	1989	1993	1995
liczba tranzystorów	130K	275K	1.2M	3.1M	5.5M
szybkość (MIPS)	1	5	20	100	150

†W.Stallings, *Organizacja i architektura systemu komputerowego*, WNT, Warszawa, 2000.

Prawo Moore'a (1965)

- Gordon Moore – założyciel i wiceprezydent firmy Intel

Sformułowanie I:

- Liczba tranzystorów, które można zmieścić na jednym calu kwadratowym płytki krzemowej podwaja się co 12 miesięcy.

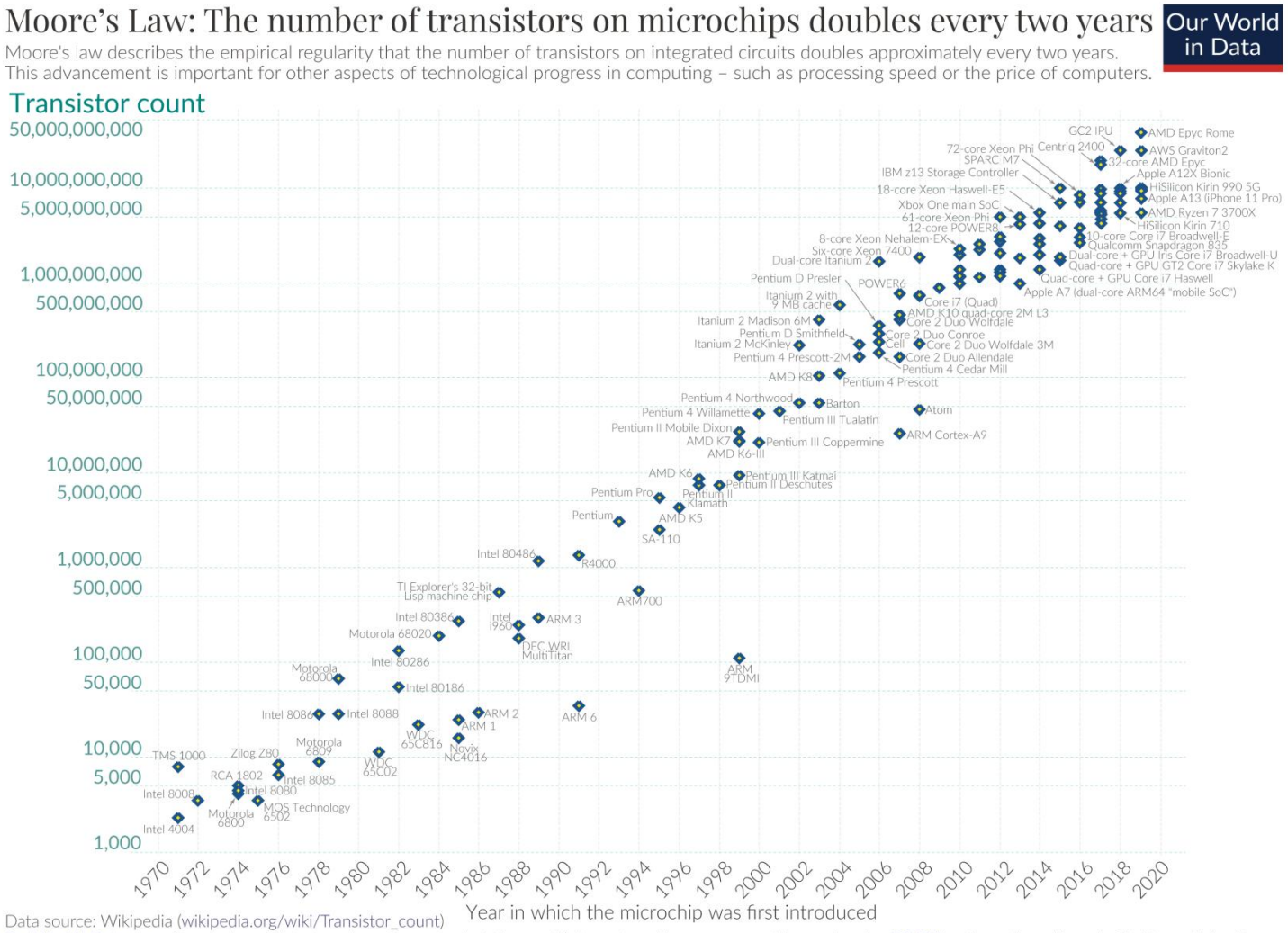
Sformułowanie II:

- Liczba tranzystorów (na jednostce powierzchni płytki krzemowej), która prowadzi do najmniejszych kosztów na jeden tranzystor, podwaja się w przybliżeniu co 12 miesięcy.

Sformułowanie III:

- Wydajność systemów komputerów ulega podwojeniu co około 18 miesięcy.

Prawo Moore'a



Architektura współczesnego procesora

- przetwarzanie potokowe
- Superskalarność
- przewidywanie rozgałęzienia (branch prediction), spekulatywne wykonywanie rozkazów
- analiza przepływu danych, tj. badanie zależności między rozkazami i wykonywanie ich nawet w kolejności innej niż w programie, aby zmniejszyć opóźnienia
- hiperwątkowość (hyper-threading)
- instrukcje SIMD (Single Instruction Multiple Data): MMX (MultiMedia Extensions), SSE (Streaming SIMD Extensions), 3DNow (3D NO Waiting)
- Wielordzeniowość
- Intel VT (vmx), AMD V (svm)
- GPGPU (General-Purpose Graphical Processor Unit)
- Cell-BE (Cell Broadband Engine)

Właściwości procesorów

	CISC		RISC		SS	
	(a)	(b)	(c)	(d)	(e)	(f)
rok powstania	1978	1989	1988	1991	1990	1989
liczba rozkazów	303	235	51	94	184	62
rozmiar rozkazu [B]	2-57	1-11	4	32	4	4,8
tryby adresowania	22	22	3	1	2	11
liczba rejestrów	16	8	32	32	32	23-256
cache [KB]	64	8	16	128	32-64	0.5

(a) VAX 11/780, (b) Intel 80486, (c) Motorola 88000 (d) MIPS R4000, (e) IBM RS 6000, (f) Intel 80960

- CISC (Complex Instruction Set Computer) komputer o pełnej liście rozkazów
- RISC (Reduced Instruction Set Computer) komputer o zredukowanej liście rozkazów
- SS - superskalarne

Klasyfikacja systemów operacyjnych

- Systemy przetwarzania bezpośredniego (ang. on-line processing systems) — **systemy interakcyjne** – występuje bezpośrednia interakcja pomiędzy użytkownikiem a systemem, – wykonywanie zadania użytkownika rozpoczyna się zaraz po jego przedłożeniu

- Systemy przetwarzania pośredniego (ang. off-line processing systems) — **systemy wsadowe** – występuje znacząca zwłoka czasowa między przedłożeniem a rozpoczęciem wykonywania zadania, – niemożliwa jest ingerencja użytkownika w wykonywanie zadania.

- **Systemy jednozadaniowe** — niedopuszczalne jest rozpoczęcie wykonywania następnego zadania użytkownika przed zakończeniem poprzedniego.

- **Systemy wielozadaniowe** — dopuszczalne jest istnienie jednocześnie wielu zadań (procesów), którym zgodnie z pewną strategią przydzielany jest procesor.

- Systemy dla **jednego** użytkownika — zasoby przeznaczone są dla jednego użytkownika (np. w przypadku komputerów osobistych), nie ma mechanizmów autoryzacji, a mechanizmy ochrony informacji są ograniczone.

- Systemy **wielodostępne** — wielu użytkowników może korzystać ze zasobów systemu komputerowego, a system operacyjny gwarantuje ich ochronę przed nieupoważnioną ingerencją.

- Systemy czasu **rzeczywistego** (ang. real-time systems) — zorientowane na przetwarzanie z uwzględnieniem czasu zakończenie zadania, tzw. linii krytycznej (ang. deadline).

- Systemy **sieciowe** i **rozproszone** (ang. network and distributed systems) — umożliwiają zarządzanie zbiorem rozproszonych jednostek przetwarzających, czyli zbiorem jednostek (komputerów), które są zintegrowane siecią komputerową i nie współdzielą fizycznie zasobów

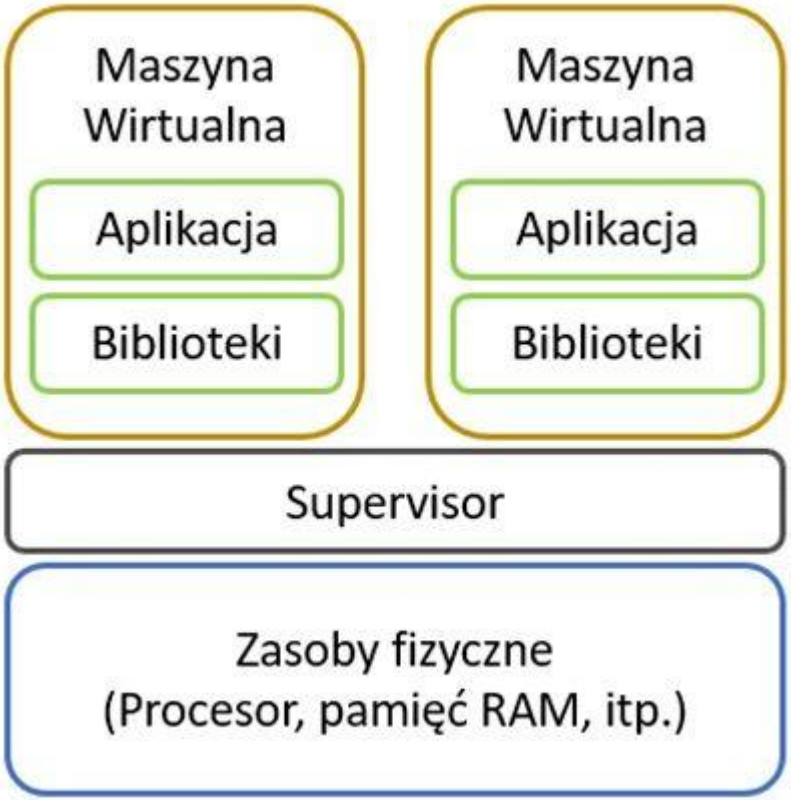
- Mobilne systemy operacyjne— tworzone dla rozwiązań typu PDA, czy telefonów komórkowych.

Wirtualizacja

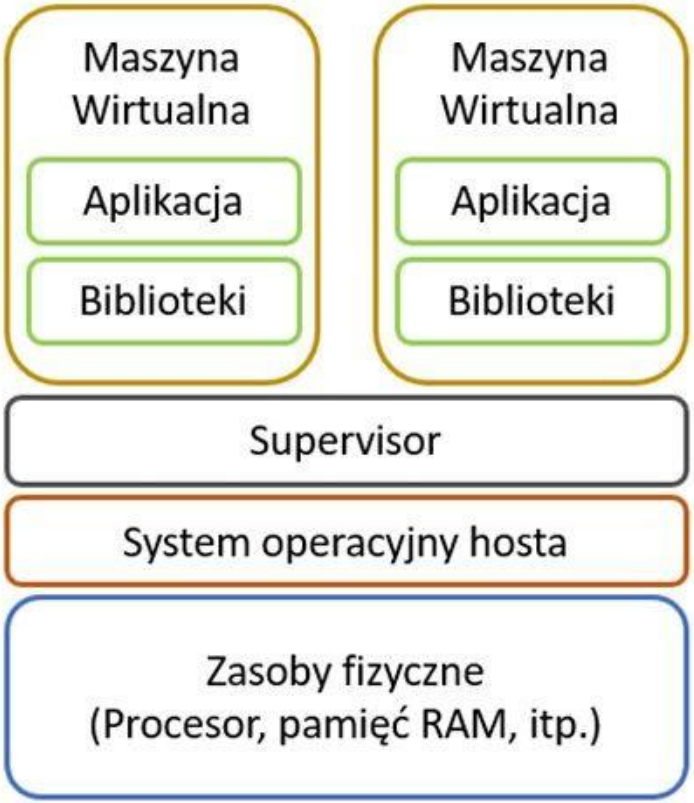
- Wirtualizacja to symulowanie systemu operacyjnego i jego zasobów fizycznych. Maszyna wirtualna zachowuje się jak oddzielny komputer, na którym możemy pracować.

- Tworzenie maszyn wirtualnych jest możliwe dzięki specjalnemu oprogramowaniu nazywane *supervisor* lub *hypervisor*, które służy do zarządzania maszynami wirtualnymi.

- Typ pierwszy – hypervisor/supervisor działa bezpośrednio na maszynie, które służy do tworzenia środowisk wirtualnych. Przykłady: Hyper-V, Xen Server



- Typ drugi – hypervisor/supervisor działa wewnątrz systemu operacyjnego hosta i poprzez ten system zarządza zasobami dla maszyn wirtualnych. Innymi słowy jest to oddzielny program działający w naszym systemie operacyjnym. Przykłady: Virtualbox, Vmware Workstation

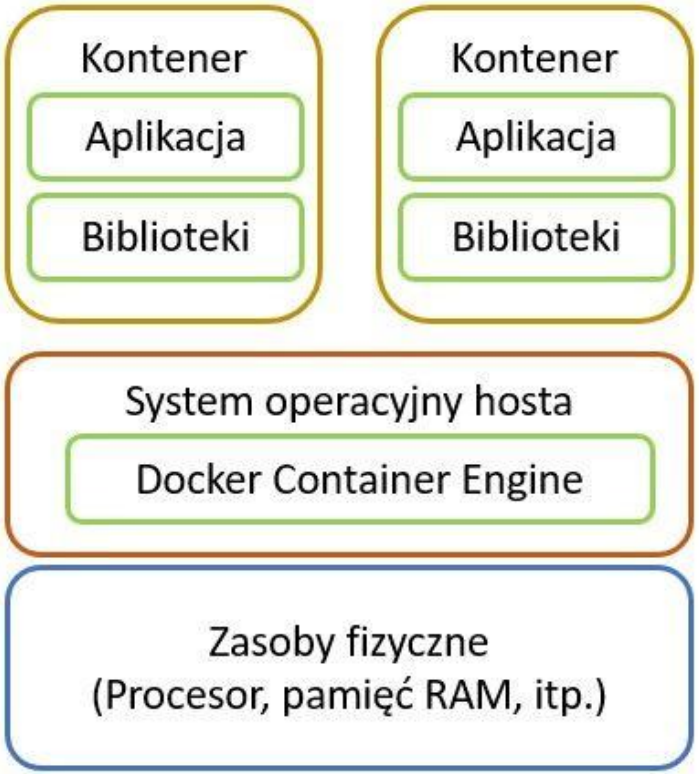


- Zalety:
  - izolowanie aplikacji
  - bezpieczeństwo
  - łatwa migracja

Wady:
  - zasobożerność
  - nieoptymalne gospodarowanie zasobami

Konteneryzacja

- Cały proces polega ona zamykaniu aplikacji, jej zależności, procesów itp.. w wirtualnej jednostce zwanej kontenerem. Z punktu widzenia aplikacji, kontenery są odrębnymi i niezależnymi od siebie środowiskami. Mimo wszystko to nadal nie jest wirtualizacja.



Procesy

- Proces jest wykonywanym programem.

- Wykonanie procesu musi przebiegać w sposób sekwencyjny (w dowolnej chwili może zostać wykonany co najwyżej jeden rozkaz)

Elementy procesu

- kod programu
- licznik rozkazów - bieżąca czynność procesu
- aktualna zawartość rejestrów procesora
- stos procesu
- sekcja danych

Stan procesu

- nowy - proces został utworzony
- aktywny - wykonywa są instrukcje
- oczekujący - czeka na wystąpienie zdarzenia, np. zakończenie operacji I/O.
- gotowy - oczekuje na przydział procesora
- zakończony - zakończył działanie

Stan procesu

Blok kontrolny procesu

Blok kontrolny procesu

- stan procesu (nowy, gotowy, aktywny itp.)
- licznik rozkazów - adres następnego rozkazu do wykonania
- rejestr procesora
- informacje do planowania przydziału procesora (priorytet, wskaźnik do kolejek)
- Informacje zarządzania pamięcią
- Informacje do rozliczeń (użyty czas procesora, czas całkowity, konta użytkowników i uprawnienia)
- Informacje o stanie I/O (urządzenia przydzielone do procesu, wykaz otwartych plików itp.)

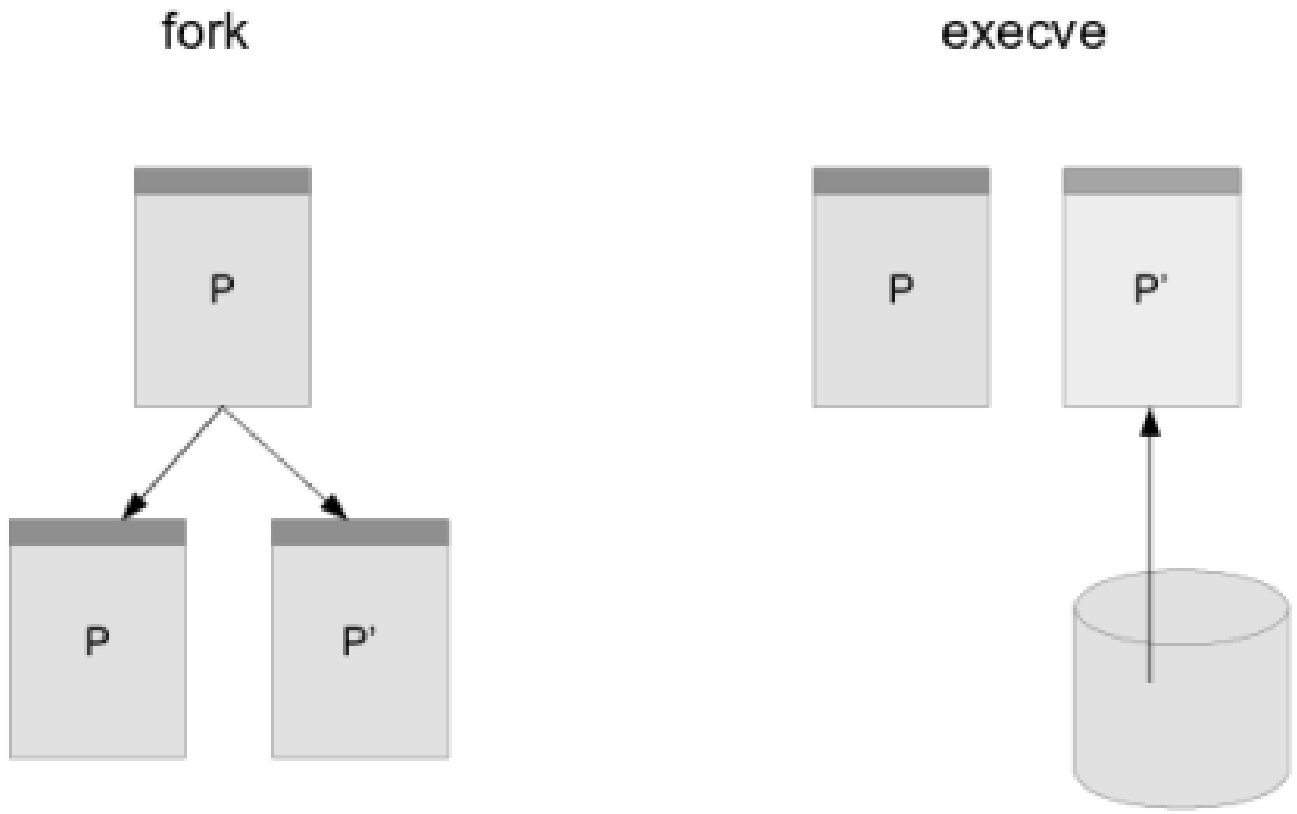
Tworzenie procesu

- Proces macierzysty tworzy potomne za pomocą funkcji systemowej.
- Nowy proces też może tworzyć procesyy potomne - powstaje wtedy drzewo procesów.
- Proces macierzysty i potomek mogą dzielić w całości, w części, lub wcale nie dzielić ze sobą zasobów.
- Proces macierzysty i potomek działają równolegle, lub też proces macierzysty czeka, aż potomek zakończy działanie.
- Proces potomny może być kopią procesu macierzystego lub otrzymać zupełnie nowy program.

Tworzenie procesu w Unixie

- Nowy proces tworzy się za pomocą funkcji systemowej **fork**.
- Potomek zawiera kopię przestrzeni adresowej przodka - daje to możliwość komunikacji pomiędzy procesami.
- Funkcja systemowa **execve** ładuje nowy program do przestrzeni adresowej procesu (niszcząc poprzednią zawartość) i rozpoczyna jego wykonanie.
- Proces macierzysty albo tworzy nowych potomków, albo czeka na zakończenie procesu potomnego.

Tworzenie procesu w Unixie



Tworzenie procesu - porównanie

Tworzenie procesu

- – POSIX: fork
- – Windows: CreateProcess

Usuwanie procesu

- – POSIX: exit, abort, kill
- – Windows: ExitProcess, TerminateProcess

Kończenie procesu

- Po zakończeniu ostatniej instrukcji proces prosi o system o usunięcie (funkcja systemowa **exit**)
- System przekazuje wynik działania potmoka do procesu macierzystego (wykonującego funkcję systemową **wait**).
- Zamyka procesowi potomnemu zasoby (pamięć, otwarte pliki, buforowanie).

Kończenie procesu

- Proces macierzysty może "awaryjnie" zakończyć proces potomny, np. gdy:

- proces potomny nadużył przydzielonych zasobów
- zadanie wykonane przez proces potomny stało się zbędne

- proces macierzysty kończy się a system nie pozwala na działanie "sieroty".

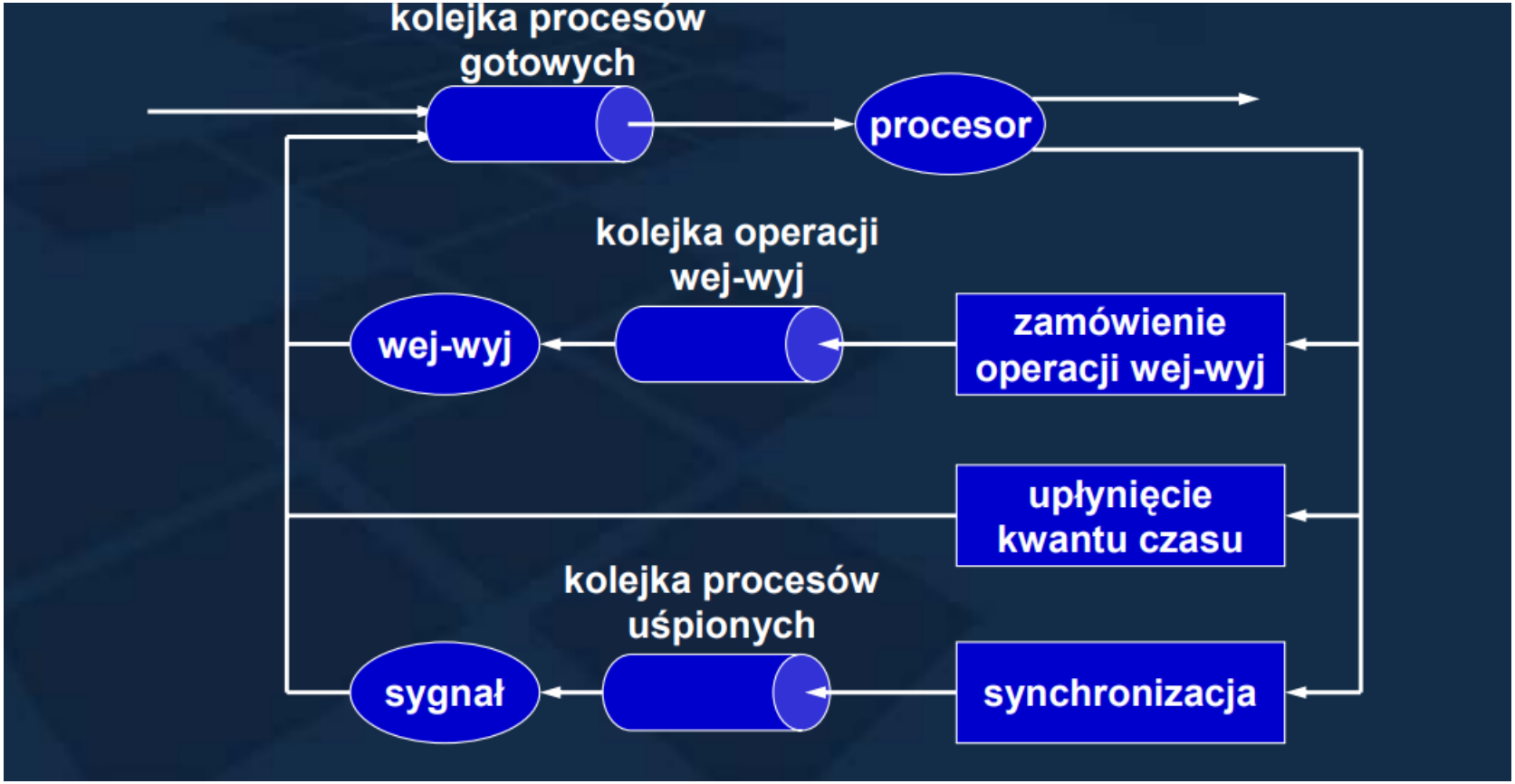
Obsługa procesów

- Zawieszanie i aktywacja procesu
- Wstrzymywanie i wznawianie procesu
- Zmiana priorytetu procesu
  – POSIX: nice (setpriority)
  – Windows: SetPriorityClass
- Oczekiwanie na zakończenie procesu
  – POSIX: wait, waitpid
  – Windows: brak bezpośredniego wsparcia, należy użyć odpowiednich mechanizmów synchronizacji

Kolejki procesów

- Kolejka zadań (ang. job queue) — wszystkie procesy systemu.
- Kolejka procesów gotowych (ang. ready queue) — procesy gotowe do działania, przebywające w pamięci głównej.
- Kolejka do urządzenia (ang. device queue) — procesy czekające na zakończenie operacji wejścia-wyjścia.
- Kolejka procesów oczekujących na sygnał synchronizacji od innych procesami (np. kolejka procesów na semaforze).

Kolejki procesów



Planista

- **Planista krótkoterminowy**, planista przydziału procesora (ang. CPU scheduler) — zajmuje się przydziałem procesora do procesów gotowych.
- **Planista średnioterminowy** (ang. medium-term scheduler) — zajmuje się wymianą procesów pomiędzy pamięcią główną a pamięcią zewnętrzną (np. dyskiem).
- **Planista długoterminowy**, planista zadań (ang. long-term scheduler, job scheduler) — zajmuje się ładowaniem nowych programów do pamięci i kontrolą liczby zadań w systemie oraz ich odpowiednim doborem w celu zrównoważenia wykorzystania zasobów.

Wątki

Wątek (lekki proces, ang. lightweight process — LWP) jest obiektem w obrębie procesu ciężkiego (heavyweight), posiadającym własne sterowanie i współdzielącym z innymi wątkami tego procesu przydzielone (procesowi) zasoby:

- – segment kodu i segment danych w pamięci
- – tablicę otwartych plików
- – tablicę sygnałów

Realizacja wątków

- Realizacja wątków na poziomie jądra systemu operacyjnego — jądro tworzy odpowiednie struktury (blok kontrolny) do utrzymywania stanu wątku.
- Realizacja wątków na poziomie użytkownika — struktury związane ze stanami wątków tworzone są w przestrzeni adresowej procesu.

Realizacja wątków na poziomie jądra

Wątek posiada własny blok kontrolny w jądrze systemu operacyjnego, obejmujący:

- stan licznika rozkazów,
- stan rejestrów procesora,
- stan rejestrów związanych z organizacją stosu.

Własności realizacji wątków na poziomie jądra:

- przełączanie kontekstu pomiędzy wątkami przez jądro,
- większy koszt przełączanie kontekstu,
- bardziej sprawiedliwy przydział czasu procesora.

Obsługa wątków

- Tworzenie wątku

– POSIX: pthread_create

– Windows: CreateThread, CreateRomoteThread

- Usuwanie wątku

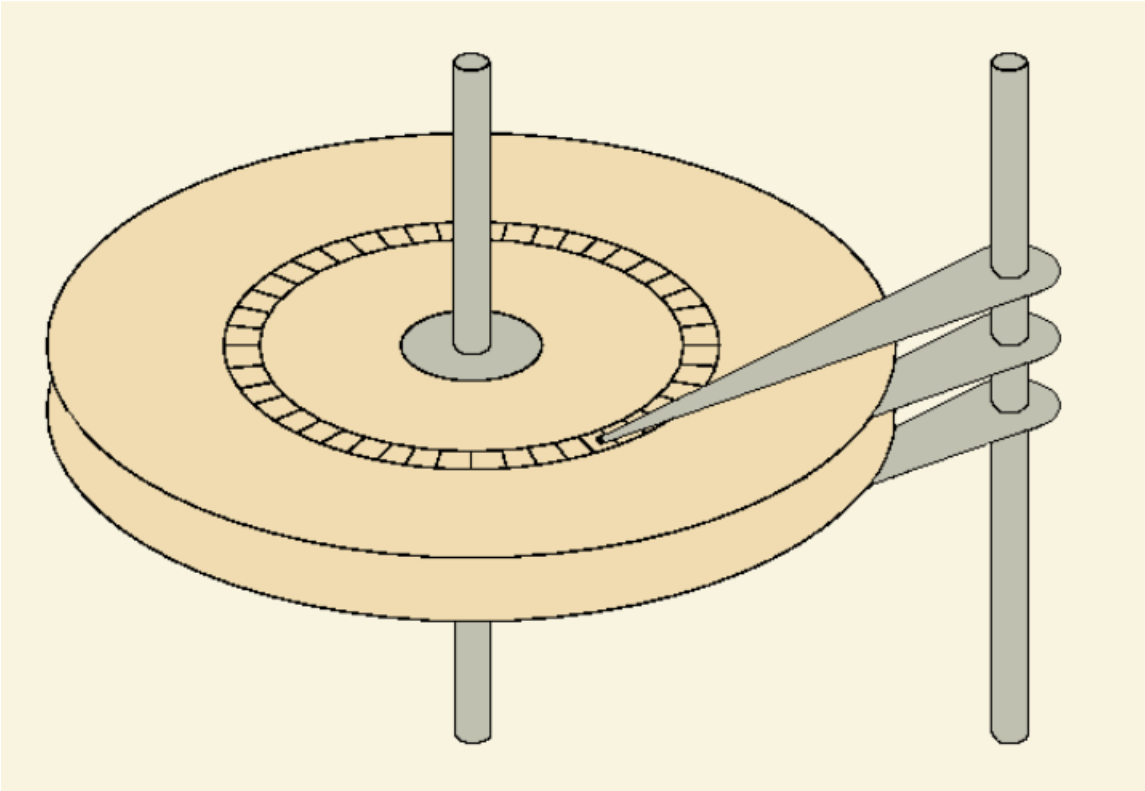
– POSIX: pthread_exit, pthread_cancel

– Windows: ExitThread, TerminateThread

Zarządzanie dyskiem

- Dyski magnetyczne stanowią najpopularniejszy rodzaj masowej pamięci trwałej.
- Dysk składa się z kilku talerzy osadzonych sztywno na wspólnej osi i wirujących z dużą prędkością. Każdy z talerzy pokryty jest z obu stron warstwą magnetyczną, na której można zapisywać informacje.

Zarządzanie dyskiem



Zarządzanie dyskiem

Zarządzanie dyskiem

- Powierzchnie talerzy są podzielone na koncentrycznie ułożone **ścieżki**.

- Ścieżki są dalej podzielone na sektory. **Sektor** to podstawowa jednostka zapisu/odczytu z dysku. Typowa wielkość sektora to 512B. Informacje są zapisywane i odczytywane przez głowice unoszące się tuż nad powierzchnią talerzy.

Zarządzanie dyskiem

- Głowice są przymocowane do ramion osadzonych sztywno na wspólnej osi. Przesuwając ramiona, przesuwamy głowice między ścieżkami. Zestaw ścieżek ze wszystkich powierzchni talerzy, tak samo odległych od osi talerzy nazywamy **cylindrem**. Wszystkie sektory w cylindrze mogą być odczytywane/zapisywane bez konieczności przesuwania głowic.

Zarządzanie dyskiem. Formatowanie.

- Powierzchnię talerzy można przyrównać do kartki papieru. Oprócz informacji przechowywanych na dysku, na talerzach zapisane są dodatkowe informacje wyznaczające podział dysku na sektory.
- Informacje te można porównać do kratek na papierze w kratkę. Proces nagrywania tych informacji organizacyjnych nazywany jest **formatowaniem**
- (**niskopoziomowym**).
- Terminu formatowanie (**wysokopoziomowe**) używa się również w odniesieniu do utworzenia na nowym dysku systemu plików.

Zarządzanie dyskiem. Formatowanie

- Fabrycznie nowe dyski są już **zwykle sformatowane** niskopoziomowo. Chcąc określić, którego sektora ma dotyczyć operacja dyskowa, musimy podać jego współrzędne: powierzchnię talerza, ścieżkę i pozycję sektora na ścieżce.
- We współczesnych napędach dyskowych fizyczne współrzędne sektorów są ukryte przed resztą systemu komputerowego. Dysk udostępnia linową logiczną tablicę sektorów (ang. **linear block addressing**). Dzięki temu, na ścieżkach zewnętrznych może być więcej sektorów niż na ścieżkach wewnętrznych i jest to zrealizowane w sposób niewidoczny dla systemu operacyjnego.

Zarządzanie dyskiem. Uszkodzenia

- Dyski magnetyczne są precyzyjnymi urządzeniami podatnymi na uszkodzenia. Jedno z takich uszkodzeń polega na **uszkodzeniu sektora**.
- Czasami takie błędy można naprawić. Informacje są zapisywane w sektorach z użyciem kodu korygującego błędy. Zapisanej informacji towarzyszy rodzaj sumy kontrolnej, która w przypadku przekłamania ograniczonej ilości informacji (1-2 bity) pozwala na wykrycie przekłamania oraz odtworzenie oryginalnej zawartości.
- Jeżeli nośnik magnetyczny uległ w danym miejscu uszkodzeniu, to mamy do czynienia z **tzw. uszkodzonym sektorem**.

Zarządzanie dyskiem. Uszkodzenia.

- Systemy plików zwykle potrafią radzić sobie z uszkodzonymi sektorami.
- Blok dyskowy zawierający taki sektor jest zaznaczany jako **uszkodzony** i nie jest więcej wykorzystywany. Jeżeli udało się odtworzyć zawartość uszkodzonego sektora, to jest ona przenoszona w inne miejsce na dysku.
- Współczesne dyski potrafią również same radzić sobie z uszkodzonymi sektorami. Każdy cylinder zawiera pewną pulę zapasowych sektorów.
- Sektory zapasowe nie są widoczne dla reszty systemu komputerowego. W przypadku wykrycia uszkodzonego sektora jest on automatycznie zastępowany przez sektor zapasowy.

Zarządzanie dyskiem. Parametry.

- O prędkości dysku decyduje kilka parametrów:
- **czas szukania** (ang. seek time) – jest to czas potrzebny na przemieszczenie ramienia z głowicami nad wybrany cylinder
- **opóźnienie obrotowe** (ang. rotation latency) – jest to czas potrzebny na to, aby wybrany sektor przejechał pod głowicami; czas ten zależy od prędkości obrotowej talerzy
- **szybkość przesyłu informacji** z/do napędu dyskowego

Zarządzanie dyskiem. Parametry.

- Opóźnienie obrotowe i szybkość przesyłu informacji zależą tylko od konstrukcji dysku.
- Czas szukania zależy również od systemu operacyjnego. Często operacje dyskowe nie dotyczą losowo wybranych sektorów dysku, lecz sektorów tworzących kolejne bloki pliku. **System operacyjny powinien starać się tak rozmieszczać pliki na dysku, aby kolejne bloki pliku były zapisane w miarę po kolei** na dysku. Wówczas często kolejna operacja dyskowa będzie odnosić się do tego cylindra, nad którym akurat znajdują się głowice.
- W ten sposób minimalizuje się czas szukania.

Zarządzanie dyskiem. Strategie szeregowania.

- Wyobraźmy sobie sytuację, w której kilka odwołań do różnych miejsc na dysku oczekuje na wykonanie.
- Czy kolejność, w której je wykonamy ma znaczenie? Okazuje się że tak. Ma ona wpływ na czas szukania na dysku. **Im większą drogę muszą przebyć głowice, tym dłuższy czas szukania.**

Przykładowe strategie

FCFS

Strategia FCFS (ang. first-come first-served) to najprostsza z możliwych strategii. Polega ona na wykonywaniu odwołań do dysku w takiej kolejności, w jakiej się one pojawiają.

Przykładowe strategie

SSTF

Strategia SSTF (ang. shortest seek time first) polega na tym, że w pierwszej kolejności obsługiwane jest to odwołanie do dysku, które jest najbliżej aktualnej pozycji głowicy. Strategia ta jest dobra, jeżeli dysk nie jest intensywnie używany przez wiele procesów. W przeciwnym przypadku jest ona podatna na zagłodzenie. Może się zdarzyć, że pewne odwołanie będzie oczekiwać na wykonanie, ale cały czas będą spływać inne odwołania, które będą bliżej aktualnej pozycji głowicy.

Przykładowe strategie

SCAN

W strategii scan głowice poruszają się wahadłowo od skrajnie zewnętrznego do skrajnie wewnętrznego cylindra i z powrotem. Po drodze zatrzymują się wykonując odwołania do napotykanych cylindrów. Strategia ta jest lepsza niż FCFS i nie występuje w niej zagłodzenie. Ma jednak pewne wady. W strategii tej średni czas oczekiwania na odwołanie do dysku zależy od miejsca na dysku. Głowice dwa razy częściej przesuwają się nad środkowymi cylindrami niż nad skrajnymi. Inna wada polega na tym, że głowice niepotrzebnie przesuwają się aż do skrajnych cylindrów.

Przykładowe strategie

Inne typy strategii:

C–SCAN

LOOK

C–LOOK

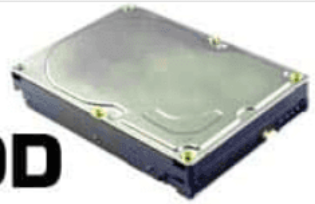
Dyski SSD


V-NAND SSD
980 PRO
PCIe 4.0 NVMe M.2
www.samsung.com/ssd
SAMSUNG ELECTRONICS CO., LTD.
1TB
SAMSUNG


Micron
2400
M.2 SSD | PCIe® NVMe™


crucial
2.5-inch SOLID STATE DRIVE

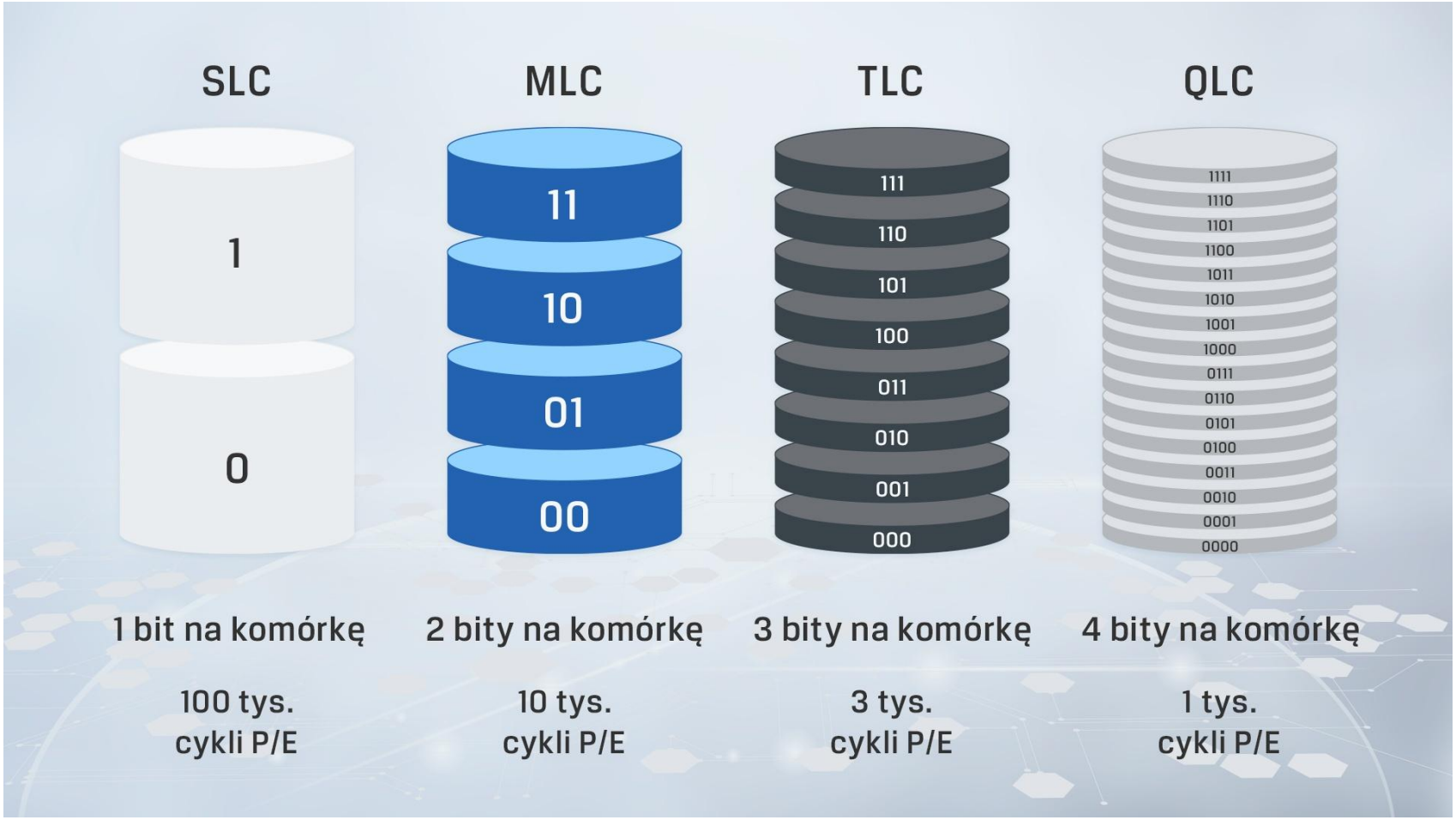
SSD vs HDD



Usually 10 000 or 15 000 rpm SAS drives

SSD		HDD
0.1 ms	**Access times** SSDs exhibit virtually no access time	5.5 ~ 8.0 ms
SSDs deliver at least 6000 io/s	**Random I/O Performance** SSDs are at least 15 times faster than HDDs	HDDs reach up to 400 io/s
SSDs have a failure rate of less than 0.5 %	**Reliability** This makes SSDs 4 - 10 times more reliable	HDD"s failure rate fluctuates between 2 ~ 5 %
SSDs consume between 2 & 5 watts	**Energy savings** This means that on a large server like ours, approximately 100 watts are saved	HDDs consume between 6 & 15 watts
SSDs have an average I/O wait of 1 %	**CPU Power** You will have an extra 6% of CPU power for other operations	HDDs' average I/O wait is about 7 %
the average service time for an I/O request while running a backup remains below 20 ms	**Input/Output request times** SSDs allow for much faster data access	the I/O request time with HDDs during backup rises up to 400~500 ms
SSD backups take about 6 hours	**Backup Rates** SSDs allows for 3 - 5 times faster backups for your data	HDD backups take up to 20~24 hours

Dyski SSD


SLC
MLC
TLC
QLC
1
0
11
10
01
00
111
110
101
100
011
010
001
000
1111
1110
1101
1100
1011
1010
1001
1000
0111
0110
0101
0100
0011
0010
0001
0000
1 bit na komórkę
2 bity na komórkę
3 bity na komórkę
4 bity na komórkę
100 tys.
cykli P/E
10 tys.
cykli P/E
3 tys.
cykli P/E
1 tys.
cykli P/E

Pamięci NAND. SLC

- **Zaleta: największa wytrzymałość – wada: wysoka cena i mała pojemność**
- Przechowuje tylko po 1 bicie informacji na komórkę. W efekcie komórka przechowuje wartość 0 lub 1, co pozwala szybciej zapisywać i odczytywać dane. Pamięć typu SLC zapewnia najlepszą wydajność i najwyższą wytrzymałość rzędu **100 000 cykli P/E** dzięki czemu może służyć dłużej niż inne rodzaje pamięci NAND. Jednak niska gęstość zapisu danych sprawia, że SLC jest najdroższym rodzajem pamięci NAND i dlatego nie jest powszechnie stosowana w urządzeniach konsumenckich. Zwykle wykorzystuje się ją w serwerach i innych zastosowaniach przemysłowych.

Pamięci NAND. MLC

- **Zaleta: niższa cena niż SLC – wada: wolniejsze działanie i mniejsza wytrzymałość niż SLC**
- Przechowuje wiele bitów na komórkę, chociaż termin MLC zwykle oznacza 2 bity. Pamięć typu MLC charakteryzuje się większą gęstością zapisu danych niż pamięć typu SLC i dlatego może być stosowana w nośnikach o większej pojemności. Pamięć typu MLC to dobre połączenie ceny, wydajności i wytrzymałości. Pamięć typu MLC jest bardziej wrażliwa na błędy w zapisie danych i przy 10 000 cyklach P/E charakteryzuje się mniejszą wytrzymałością niż pamięć typu SLC. Pamięć typu MLC stosuje się zwykle w urządzeniach konsumenckich, w których wytrzymałość odgrywa mniejsze znaczenie.

Pamięci NAND. TLC

- **Zaleta: najniższa cena i duże pojemności – wada: mała wytrzymałość**

- Pamięć NAND z komórkami trójpoziomowymi (TLC) przechowuje po 3 bity informacji na komórkę. Większa liczba bitów na komórkę pozwala zmniejszyć koszt i zwiększyć pojemność pamięci. Ma to jednak negatywny wpływ na wydajność i wytrzymałość, która wynosi tylko 3000 cykli P/E. Pamięć typu TLC wykorzystuje się w wielu urządzeniach konsumenckich, ponieważ jest to najtańsze rozwiązanie.

Pamięci NAND. 3D NAND

- **Zaleta: najniższa cena i duże pojemności – wada: mała wytrzymałość**

- Rozmieszczenie komórek pionowo a nie poziomo jak w innych pamięciach. Większa gęstość pamięci zapewnia większą pojemność bez znacznego wzrostu ceny. Pamięć 3D NAND charakteryzuje się również większą wytrzymałością i mniejszym zużyciem energii.
- Ogólnie rzecz biorąc, NAND jest niezwykle ważną technologią pamięci, ponieważ zapewnia krótki czas kasowania i zapisu przy niższym koszcie na bit.

System plików

- Plik jest logiczną jednostką magazynowania informacji w pamięci nieulotnej.
- Plik jest nazwanym zbiorem powiązanych ze sobą informacji zapisanym w pamięci pomocniczej.
- Plik jest ciągiem bitów, bajtów, wierszy lub rekordów.

Atrybuty pliku

- Atrybuty pliku:

- Nazwa (zgodna z regułami dla danego systemu operacyjnego),
- Typ (jeżeli system tego wymaga),
- Położenie (wskaźnik do urządzenia i położenie na tym urządzeniu),
- Rozmiar (w bajtach, słowach lub blokach),
- Ochrona (prawa dostępu),
- Czas, data, identyfikator właściciela,
- Ewentualnie inne metadane

Historia

- Uniksowe mainframe: Hierarchiczny system plików.
- IBM PC (DOS 1.x) - prosta lista plików na dyskietce.
- IBM PC (DOS 2.x) – katalogi, dyski twarde do 20-30MB (lub więcej ze sterownikami), kwestia systemu plików na większych dyskach nie jest ustandaryzowana.
- DOS 3.x-6.x - FAT - pełna obsługa zagnieżdżonych katalogów (z pewnymi limitami) i wielu partycji.
- Windows 95 - VFAT i 95 OSR2 - FAT32 – większe dyski, rozszerzenie FATa tak, by plik miał nazwę dłuższą niż 8+3.
- Windows NT - NTFS - w systemie plików pojawiają się strumienie, rozszerzone atrybuty, sparse files, obiekty.

Operacje na plikach

- **Tworzenie pliku**

- znalezienie miejsca w systemie plików

- wpis do katalogu

Zapisywanie pliku - podaje się nazwę (identyfikator) pliku i informację do zapisania, istotne jest miejsce od którego piszemy (wskaźnik położenia).

Czytanie pliku podaje się nazwę pliku i bufor w pamięci. Można wykorzystać ten sam wskaźnik położenia.

Zmiana pozycji w pliku - modyfikacja wskaźnika położenia.

Usuwanie pliku - zwalnia się przestrzeń zajmowaną przez

- plik i likwiduje się wpis katalogowy.

Skracanie pliku - likwidowanie części albo całej zawartości pliku bez kasowania jego nazwy i atrybutów.

Operacje na plikach

Dopisywanie - dopisywanie nowych informacji na końcu

Przemianowanie pliku - zmiana nazwy pliku, często tą samą komendą wykonuje się przesuwanie pliku, czyli zmianę jego położenia - do innego katalogu, na inny dysk.

Otwieranie pliku stosowane w wielu systemach w celu uniknięcia wielokrotnego czytania informacji o pliku - dane z katalogu kopiowane są do tablicy otwartych plików.

Zamykanie pliku - kiedy plik przestaje być potrzebny, usuwa się wpis z tablicy otwartych plików

Otwieranie i zamykanie plików w systemach wieloużytkownikowych musi uwzględniać równoczesne korzystanie z pliku przez kilka procesów

Typy plików

System rozpoznaje typy plików poprzez:

Rozszerzenia - w MSDOS niektóre typy plików określane przez rozszerzenia nazwy(*.com, *.exe...),

Liczby magiczne - oraz typowe fragmenty początku pliku - identyfikacja w systemie Unix (komenda file, plik /etc/magic),

Atrybut twórcy (w MAC OS) - czyli nazwę programu, przy pomocy którego utworzono plik.

Katalogi

Jednopoziomowy - ograniczeniem jest konieczność spełnienia warunku niepowtarzalności nazw.

Dwupoziomowy - każdy użytkownik ma własny katalog macierzysty, a w nim pliki.

Wielopoziomowe drzewiaste.

Acykliczne grafy - do pliku można dojść wieloma drogami

Ochrona plików

Można kontrolować wiele operacji:

- **czytanie pliku**
- **pisanie do pliku**, lub zapisywanie go na nowo
- **wykonywanie** - załadowanie pliku do pamięci i wykonanie go
- **dopisywanie** danych na końcu pliku
- **usuwanie** pliku i zwalnianie obszaru przez niego zajętego
- **opisywanie** - wyprowadzenie nazwy i atrybutów pliku

Ochrona plików

Klasy użytkowników pliku:

- **właściciel**

- **użytkownik**, który utworzył dany plik grupa użytkowników, którzy wspólnie korzystają z pliku i potrzebują podobnego zakresu dostępu wszyscy inni.

Poprawa wydajności dysków

pamięć podręczna - przechowywanie całych ścieżek dysku w pamięci - prawdopodobnie będą z nich w niedługim czasie czytane dane.

Wykorzystana do tego celu specjalna pamięć, lub nieużywana pamięć główna.

wczesne zwalnianie - usuwanie bloku z bufora natychmiast, gdy pojawia się zamówienia na następny (oszczędza pamięć)

czytanie z wyprzedzeniem - z zamówionym blokiem czyta się kilka nastepnych, gdyż prawdopodobnie zaraz będą potrzebne.

RAM-dysk - wszystkie operacje dyskowe przeprowadza się w pamięci.

Zawartość RAM-dysku jest pod kontrolą użytkownika. Wada – zawartość ginie po wyłączeniu zasilania, awarii.

Opóźniony zapis - zapis z pamięci podręcznej następuje później, preferuje się nawet zapis na dysk dopiero wtedy gdy potrzeba te dane odczytać. Dzięki temu opóźnia się zbędne zapisy danych tymczasowych.

Czynności operatorskie

Sprawdzanie spójności - po awarii systemu, czy np po

wyłączeniu „z kontaktu". Program chkdsk (Windows), scandisk (DOS), fsck (UNIX). Zazwyczaj uruchamiają się automatycznie (znacznik "czystości" systemu plików)

Składowanie i odtwarzanie - robienie kopii systemu plików na innym nośniku i odtwarzanie po awarii.

Kopia zapasowa, Norton Ghost (DOS, Windows), tar, backup, restore (Unix).

Składowanie przyrostowe - kopia całego systemu raz, a potem zapisywanie tylko zmienionych plików.

Kopie "wieczyste" - co jakiś czas, taśmy pozostają w archiwum "na zawsze".

System plików FAT

Katalog - zawiera: nazwy plików (8 znaków), rozszerzenie (3 znaki), długość pliku, atrybuty (h s ra), datę utworzenia pliku, wskazanie pozycji FAT

FAT (file allocation table) - tablica, której elementy odpowiadają kolejnym jednostkom alokacji (sektorom, blokom, klastrom). FAT jest umieszczony na początku dysku, w dwóch kopiach.

Przykład użycia: Plik Z1 zajmuje bloki: 7,8,11,3, Z2 zajmuje blok 4, a Z3 jest w blokach: 1,2,5,6,9

Nr:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	15	17	18	19	20
	02	05	ff	ff	06	09	08	11	ff	00	03	00	00	00	00	00	00	00	00	00

FAT 12 - każda pozycja miała 12 bitów, czyli mogła zawierać wartość 0-4096, co przy klasterach 8 KB dawało 32 MB

Kolejne wersje systemu - FAT 16 (do 2 GB) i FAT 32.

Bariera - czas przeszukiwania tablicy FAT.

System plików NTFS

Wolumin (volume) podstawowa jednostka dyskowa - może to być część dysku, cały dysk lub kilka dysków razem

Klaster (grono, cluster) - podstawowa jednostka przydziału, jest to grupa sąsiadujących sektorów. Są znacznie mniejsze niż w systemach z FAT - 4 kB dla dużych dysków. Jako adresy dyskowe używane są logiczne numery klastrów.

Plik jest obiektem strukturalnym złożonym z atrybutów.

Atrybuty pliku są strumieniami bitów. Jednym z atrybutów są też dane pliku.

Główna tablica plików (MFT) - przechowuje opisy plików, zawarte w jednym lub kilku rekordach dla każdego pliku.

Odsyłacz do pliku - niepowtarzalny identyfikator pliku, składa się z 48-bitowego numeru pliku (pozycja w MFT) i 12- bitowego numeru kolejnego

System plików NTFS

Kopia MFT - kopia pierwszych 16 pozycji MFT - do działań naprawczych,

Plik dziennika - zawiera wszystkie zmiany danych w systemie plików,

Plik woluminu - dane woluminu, dane o wersji NTFS, który go sformatował, bit informujący o konieczności chkdsk,

Tablica definicji atrybutów - typy atrybutów i dopuszczalne dla nich operacje,

Katalog główny

Plik mapy bitów - wskazuje zajęte i wolne klastry,

Plik rozruchowy - kod początkowy NT,

Plik uszkodzonych klastrów,

Usuwanie skutków awarii jest stosunkowo proste, gdyż operacje dyskowe odbywają się na zasadzie transakcji.

System plików NTFS

Każda operacja jest wykonywana na zasadzie **transakcji**.

Najpierw zmiany dokonywane na metadanych zostają zapisane w **dzienniku**, dopiero potem faktyczna operacja ma miejsce i jest zatwierdzana.

W przypadku **awarii** systemu, uruchamiane jest odzyskiwanie systemu plików.

Przeglądany jest dziennik i wszystkie niedokończone transakcje są usuwane bądź realizowane zgodnie z zapamiętanymi tam informacjami.

Takie rozwiązanie gwarantuje stabilność systemu plików, jednak czasami można utracić część informacji, które były modyfikowane w trakcie awarii, bądź znajdowały się na zepsutym klastrze.

System plików NTFS

W przypadku gdy system plików odkryje niedziałające klastry na dysku, automatycznie oznacza je jako zepsute i podstawia na jego miejsce nieużywany klaster.

W przypadku odwołania do adresu podmienionego klastra, NTFS realizuje żądanie na nowym zmapowanym klastrze.

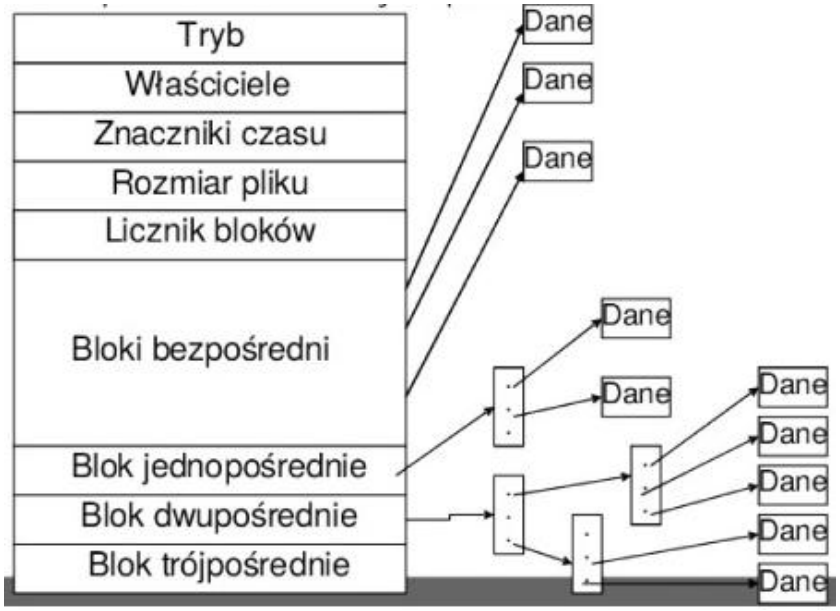
W przypadku odczytu z błędnego klastra traci się informacje na nim zawarta, jednak w przypadku zapisu, użytkownik nie dowie się nawet, że wystąpił po drodze jakiś błąd.

Powtarzające się awarie klastrów są wskazówką do wymiany nośnika na nowy.

System plików Unix

Katalog - zawiera: nazwy plików (w starszych do 256 znaków), wskazanie Inode (węzła). Za wyjątkiem znacznika, nie różni się od pliku.

I-node - zawiera pozostałe informacje o pliku,



System plików EXT4

Domyślny system plików większości dystrybucji systemu Linux.

Następca Ext3, Ext2, Ext, Minix FS...

• Możliwości:

- Urządzenia do 1EB (Eksabajt),

- Nielimitowana głębokość ścieżek katalogów,

- Dziennik transakcji z obsługą sum kontrolnych,

- API szyfrowania w trakcie operacji dyskowych,

- Opóźnienie czyszczenia i-nodeów do momentu gdy zachodzi potrzeba ich zapisu.

- Opóźnienie problemu roku 2038 o dwa bity (do roku 2446).