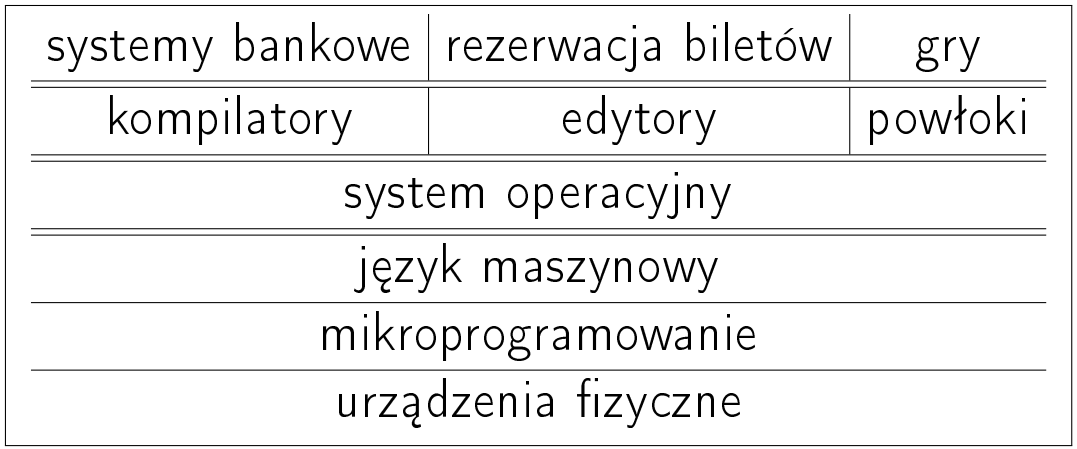
Elementy systemu komputerowego

- **sprzęt** - fizyczne komponenty komputera (procesor, pamięć, urządzenia I/O)
- **system operacyjny** - zarządza zasobami systemu komputerowego i udostępnia je programom użytkowym
- **programy użytkowe** (kompilatory, edytory, systemy baz danych, gry) - wykonują zadania dla użytkowników korzystając z przydzielonych zasobów
- **użytkownicy** (ludzie, maszyny, inne komputery), korzystają z systemu komputerowego (użytkownicy, administratorzy, programiści)

System komputerowy wg Tanenbauma



System komputerowy

Poziomy abstrakcji współczesnych systemów komputerowych
(wg Null i Lobur)

6	użytkownik	programy wykonywalne
5	języki wysokiego poziomu	C, C++, Java, FORTRAN, ...
4	asembler	kod asemblera
3	oprogramowanie systemowe	system operacyjny, biblioteki
2	maszyna	architektura zbioru rozkazów
1	sterowanie	mikrokod lub skonfigurowane sprzętowo
0	logika cyfrowa	obwody, bramki, itd.

Co to jest system operacyjny?

- System operacyjny to **oprogramowanie**, które zarządza zasobami komputera i udostępnia programom usługi oraz abstrakcje sprzętu.
- Działa jako pośrednik między użytkownikiem komputera a sprzętem komputerowym
- System operacyjny **nadzoruje i koordynuje** przydział procesora, pamięci i urządzeń wejścia-wyjścia między programami.
- System operacyjny jest odpowiedzialny za
  - **zarządzanie zasobami** komputera
  - udostępnianie programom **warstwy abstrakcji** sprzętu, które ukrywają szczegóły jego obsługi

Jądro systemu operacyjnego

- **Jądro (*kernel*)** – podstawowa część systemu operacyjnego, która działa nieustannie;
- Jądro działa w trybie uprzywilejowanym (zwanym także **trybem jądra**), który pozwala mu na wykonywanie wszystkich instrukcji procesora i pełny dostęp do sprzętu
- wszystkie pozostałe programy są programami użytkowymi, działają w **trybie użytkownika** i mają ograniczony dostęp do sprzętu

Cechy dobrego systemu operacyjnego

- **funkcjonalność** – system powinien zapewniać szeroki zakres usług dla użytkowników i programów
- **wydajność** – efektywne wykorzystanie zasobów, procesor powinien pracować dla użytkownika, minimalizacja opóźnień
- **skalowalność** – możliwość zwiększenia obciążenia i rozbudowy sprzętu bez utraty jakości działania
- **niezawodność** – wysoka dostępność, odporność na awarie, tzw. *five nines* (99.999% dostępności)
- **bezpieczeństwo** – ochrona danych i zasobów przed nieautoryzowanym dostępem, zapewnienie prywatności i integralności
- **łatwość korzystania i zarządzania** – intuicyjny interfejs, proste zarządzanie systemem i jego konfiguracją

Generacje komputerów

generacja	lata	technologia	szybkość (operacji/sek)
1	1946-1957	lampa próżniowa	40 000
2	1958-1964	tranzystor	200 000
3	1965-1971	mała i średnia skala scalenia	1 000 000
4	1972-1977	duża skala scalenia	10 000 000
5	1978-	bardzo duża skala scalenia	$>$ 100 000 000

SSI (*Small Scale Integration*) mała skala integracji, 10-100 elem./układ

MSI (*Medium Scale Integration*) średnia skala integracji, 100-1000 elem./układ

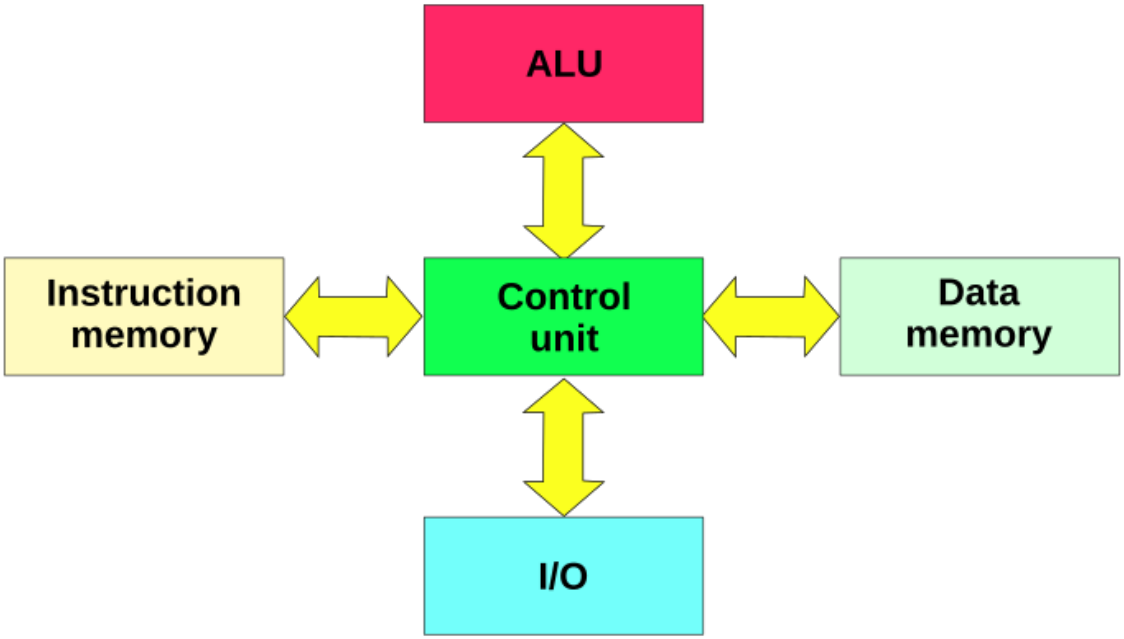
LSI (*Large Scale Integration*) duża skala integracji, 1000-10000 elem./układ

VLSI (*Very Large Scale Integration*) bardzo duża skala integracji, powyżej 10000 elem./układ

Historia rozwoju komputerów i systemów operacyjnych

- 1642-1945 (0. generacja): mechaniczne maszyny liczące
  - maszyna analityczna Babbage'a (1835)
  - ASCC (*Automatic Sequence Controlled Calculator*), Harvard Mark I, Howard Aiken, IBM (1944) - architektura harwardzka
- 1945-1953 (1. generacja): komputery na lampach próżniowych
  - bezpośrednie programowanie w języku maszynowym
  - przeprowadzanie wyłącznie obliczeń numerycznych
  - zastosowanie kart dziurkowanych do wprowadzania danych (początek lat 1950)
- brak systemów operacyjnych i języków programowania

Architektura Harwardzka



Osobna pamięć i magistrale dla danych i instrukcji

https://en.wikipedia.org/wiki/Harvard_architecture

Architektura Harwardzka

Zalety:

- duża przepustowość: równoległy dostęp do danych i instrukcji
- niezależna organizacja pamięci instrukcji i danych
- możliwość doboru różnych parametrów pamięci instrukcji i danych

Wady:

- większa złożoność i koszt implementacji
- mniejsza elastyczność i trudniejsze współdzielenie pamięci przez dane i instrukcje

Przykłady komputerów 1. generacji

- **EDVAC** *Electronic Discrete Variable Computer* (1946, IAS[1])
  architektura von Neumanna
  pamięć główna przechowująca dane i rozkazy
  każdy element pamięci ma unikalny identyfikator - adres

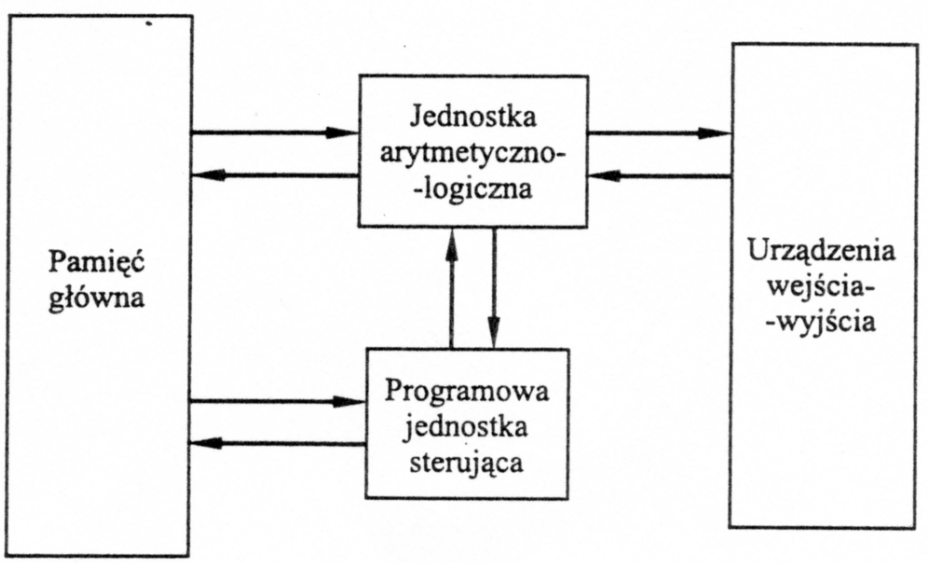
  1000 słów 40 bitowych (po 2 instrukcje (20-to bitowe) w słowie)

Pierwsze komputery komercyjne:

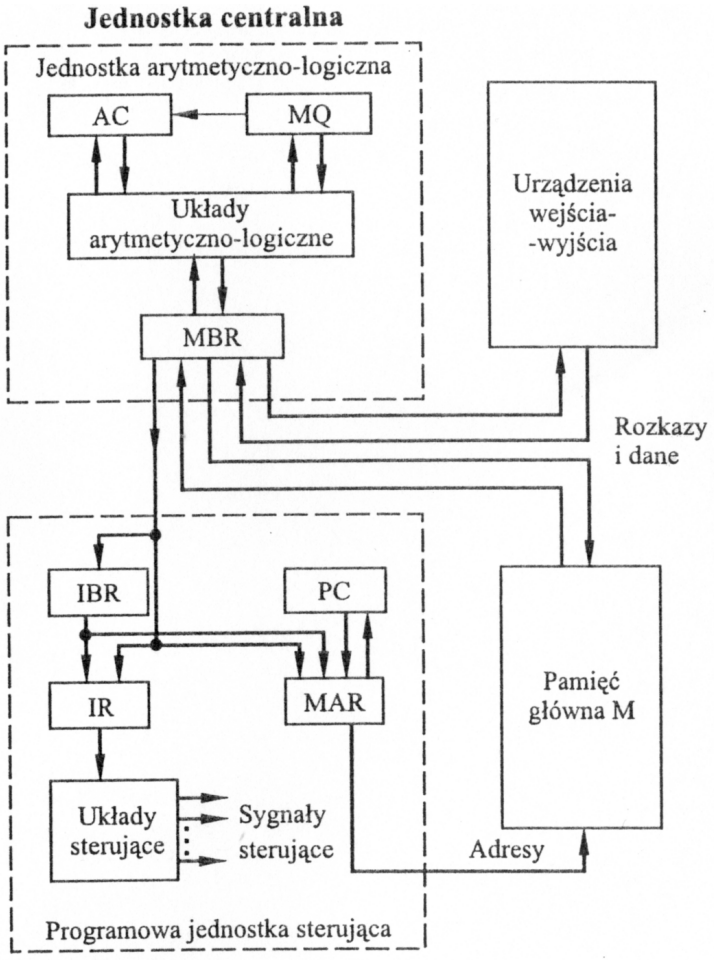
- **IBM 701** (1952) z przechowywanym programem do zastosowań naukowych, model 702 (1955) do zastosowań biznesowych

[1] Institute for Advanced Study, Princeton

Maszyna von Neumanna



- jednostka arytmetyczno-logiczna (ALU, *Arithmetic-Logic Unit*) wykonująca działania na danych binarnych
- jednostka sterująca interpretuje rozkazy z pamięci i powoduje ich wykonanie
- urządzenia wejścia-wyjścia, których pracą kieruje jednostka centralna



RYSUNEK 2.3. Struktura komputera IAS

- **rejestr buforowy pamięci** (MBR, *memory buffer register*) – przechowuje słowo odczytane z pamięci lub przygotowane do zapisu w pamięci
- **rejestr adresowy pamięci** (MAR,*memory address register*) – adres w pamięci słowa, które ma być zapisane/odczytane z MBR
- **rejestr rozkazów** (IR, *instruction register*) – 8-bitowy kod operacyjny rozkazu, który jest wykonywany
- **buforowy rejestr rozkazów** (IBR, *instruction buffer register*) – przechowuje drugi rozkaz z pobranego słowa podczas wykonywania pierwszego
- **licznik programu** (PC, *program counter*) – adres słowa pamięci z kolejną parą rozkazów
- **akumulator** (AC, *accumulator*) oraz **rejestr mnożenia i dzielenia** (MQ, *multiplier-quotient*) – czasowe przechowywanie argumentów i wyników operacji prowadzonych przez ALU.

Cykle komputera IAS:

- **cykl pobrania rozkazu:** kod operacji następnego rozkazu jest ładowany do IBR, a część adresowa – do MAR
- **cykl wykonania:** układy sterujące rozpoznają kod operacji i wykonują rozkaz, wysyłając odpowiednie sygnały sterujące, które powodują, że przenoszone są dane lub ALU wykonuje operację

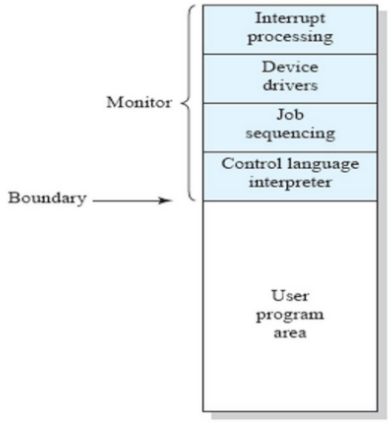
  Komputer IAS ma 21 rozkazów:

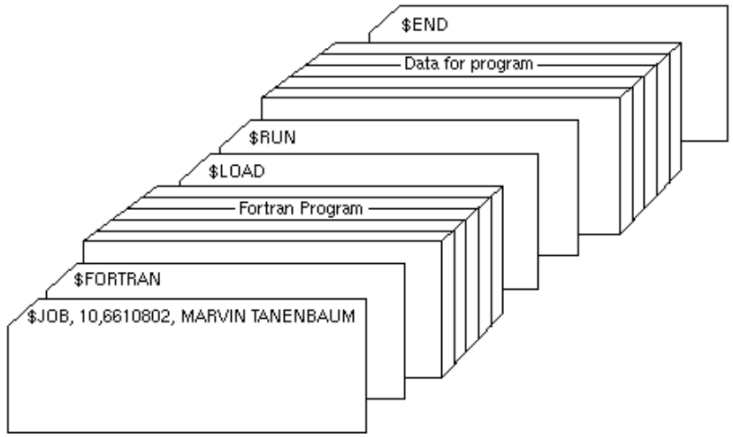
  – przenoszenie danych
  – rozgałęzienie bezwarunkowe
  – rozgałęzienie warunkowe
  – arytmetyka
  – modyfikowanie adresu

2. generacja: tranzystory i systemy wsadowe (1955-1965)

- zwiększona niezawodność
- bardziej złożone jednostki ALU oraz sterujące
- zadania wykonywane po jednym na raz
- rozgraniczenie roli operatora i programisty
- rozwój języków programowania wysokiego poziomu (np. Fortran)
- system wsadowy i praca pośrednia (wczytywanie i drukowanie *off-line*)
- pojawia się oprogramowanie systemowe: proste **monitory** (np. FSM, *Fortran Monitor System*, IBSYS) i **karty sterowania zadaniami** (poprzednik powłoki i obecnych CLI)
- **monitor** wraz z językiem kontroli zadań (JCL, *Job Control Language*) umożliwiał **przetwarzanie wsadowe** (*batch processing*), czyli automatyczne wykonywanie zadań bez interwencji użytkownika

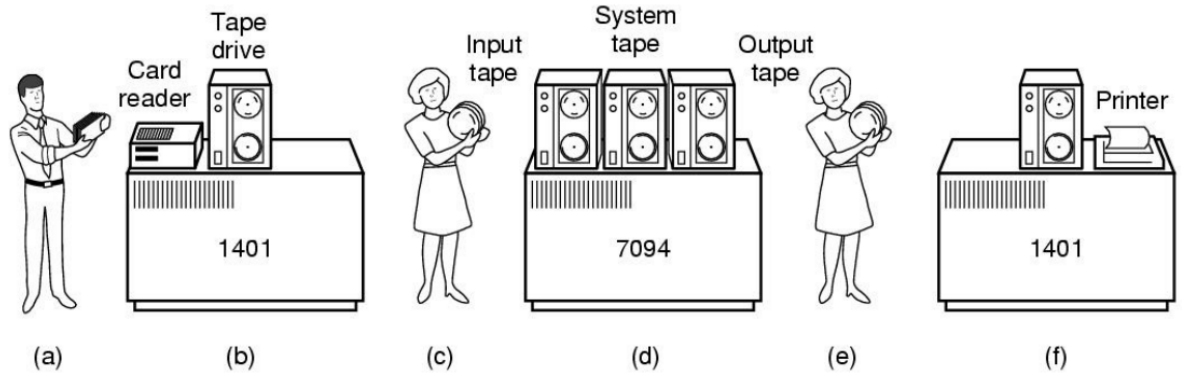
Monitor rezydujący w pamięci





- specjalne karty sterujące, instrukcje monitora
- monitor i program rezydują jednocześnie w wydzielonych obszarach pamięci
- użytkownik nie ma bezpośredniego dostępu do procesora
- czytniki kart i drukarki powolne względem jednostki obliczeniowej

Wczesne systemy przetwarzania wsadowego



- **Praca pośrednia** (*off-line*) - przetwarzanie danych, gdzie urządzenia wejścia/wyjścia (np. czytniki kart, drukarki, taśmy magnetyczne) nie są bezpośrednio sterowane przez procesor komputera, lecz przez odrębne urządzenia lub programy pośredniczące
- dane są najpierw zbierane lub przygotowywane przez mniejsze, wyspecjalizowane urządzenia, a następnie przekazywane do komputera w blokach (np. na taśmie magnetycznej)
- procesor nie traci czasu na oczekiwanie na wolne urządzenia we/wy

3. generacja: wieloprogramowość (1965-1980)

- rodzina komputerów o ujednoliconej architekturze do ogólnych zastosowań, IBM 360 (1964), ujednolicony system operacyjny (OS/360), przenośny (w teorii), przetwarzanie wsadowe
- **wieloprogramowość** - możliwość przechowywania w pamięci kilku programów jednocześnie i przełączania się między nimi
- **spooling** - kolejkowanie zadań wejścia/wyjścia w pamięci masowej. Specjalny program systemowy (*spooler*) pobiera dane z dysku i przesyła je do urządzenia w tle, we własnym, wolnym tempie. Wyeliminowało to małe komputery do obsługi kart
- pojawiają się **systemy z podziałem czasu**:
  - interaktywny dostęp wielu użytkowników, współbieżne działanie wielu programów, ochrona pamięci, systemy plików,
  - *Compatible Time-Sharing System* (CTSS)
  - *Multiplexed Information and Computing Service* (MULTICS)

4. generacja (1980-): komputery osobiste i sieci

- stacje robocze, komputery osobiste (architektura minikomputerów),
- graficzne interfejsy użytkownika (GUI)
- sieciowe systemy operacyjne
- rozproszone systemy operacyjne (układy masywnie równoległe)
- wirtualizacja

5. generacja: systemy mobilne (1990-)

- telefony łączące funkcje telefonu i PDA (*Personal Digital Assistant*), np. Nokia 9000 Communicator (1996)
- systemy operacyjne urządzeń mobilnych: Symbian OS, Windows Mobile, Palm OS, iOS, Android
- obecnie: tablety, smartfony, smartwatche, smartTV, IoT
- systemy wbudowane: RTOS, TinyOS, Contiki

Początki komputeryzacji w Polsce

- Pierwsze polskie komputery (lata 50. i 60. XX wieku)
  - Analogowy Analizator Równań Algebraicznych Liniowych (ARAL, 1954)
  - Analogowy Analizator Równań Różniczkowych (ARR), Elektroniczna Maszyna Automatycznie Licząca (EMAL)
  - XYZ (1958) wzorowany na IBM 701, pamięć naddźwiękową, 800 operacji/sek.
- System 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)[2]

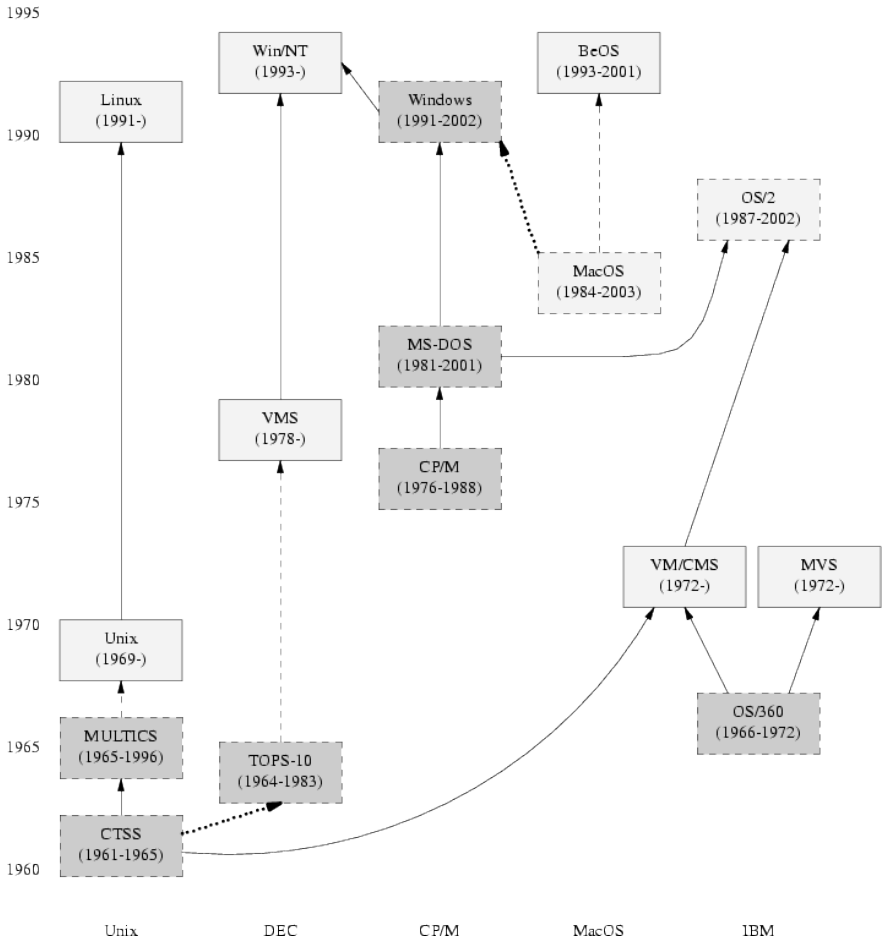
[2] Zob. System Mera 400

Systemy operacyjne z początków komputeryzacji w Polsce

- SOK-1 - tylko jeden program (komputer K-202)
- SODA System Operacyjny Dwu Aktywny (Odra) - dwa programy
- Egzekutor RTX (*Real Time Executive*), Mera 300
- Ładowacz (Mera 305)
- MSO 300 (Mera 306), system ogólnego przeznaczenia
- SOWA System Operacyjny Wielo Aktywny - potem nazwany CROOK
- CROOK 1-5, komputery K-202 i Mera 400, uniwersalny, wielozadaniowy, wielodostępowy system operacyjny z hierarchicznym systemem plików[3]

[3] Zob. https://mera400.pl/CROOK

Historia systemów operacyjnych wg E.S.Raymonda[4]

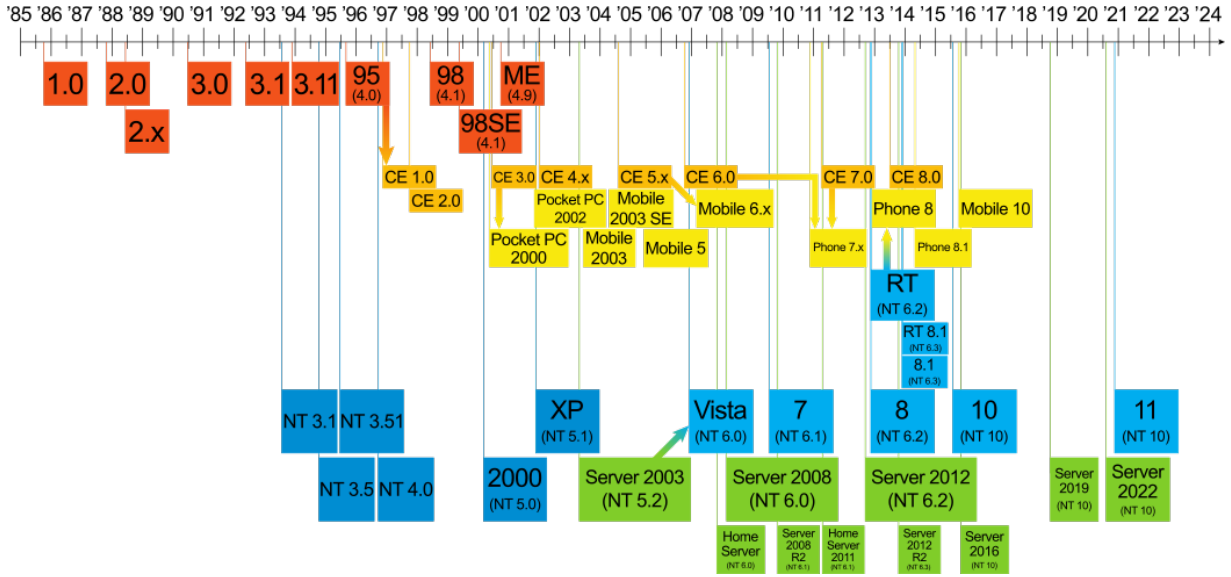


[4] *UNIX Sztuka programowania*, Wydawnictwo Helion, Gliwice, 2004. Także http://en.wikipedia.org/wiki/Timeline_of_operating_systems

MS-DOS, Windows, Mac OS

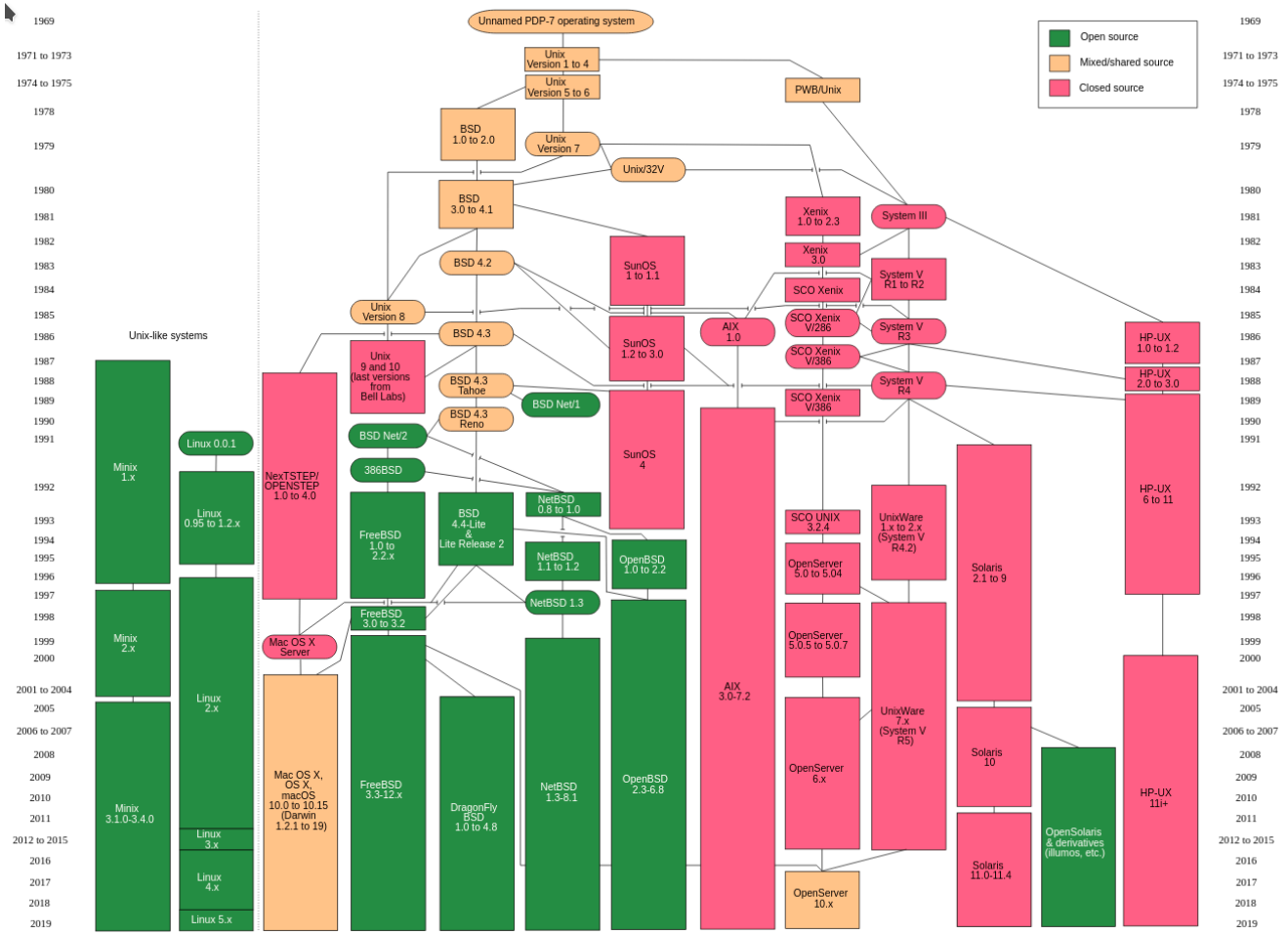
- 1974 pierwszy 8-mio bitowy procesor 8080 ogólnego przeznaczenia,
- system CP/M (*Control Program for Microcomputers*), rozwijany przez Digital Research pod kierunkiem Gary'ego Kildalla, szeroko stosowany w mikrokomputerach lat 70.
- 1981 IBM PC debiutuje z systemem PC DOS; Microsoft rozwijał jego wersję dla innych komputerów jako MS-DOS
- 1984 IBM PC/AT umacnia pozycję MS-DOS
- 1984 Apple wprowadza komputer Macintosh z GUI i myszą; wcześniej graficzne interfejsy rozwijano m.in. w Xerox PARC
- 1985 Windows 1.0 - początkowo graficzna nakładka na MS-DOS
- 1995 Windows 95: system z graficznym interfejsem, zachowujący obsługę MS-DOS

- Windows NT (*New Technology*): odrębna linia 32-bitowa systemów Windows z własnym jądrem, bezpieczna, z pełnym wywłaszczaniem procesów i ochroną pamięci.
- od 2000 dwie linie:[5]
  - serwerowa: Windows Server 2003, Windows Server 2008, ...
  - kliencka: oparte o Windows XP, Windows 7, 8, 10, 11, ...



[5] https://pl.wikipedia.org/wiki/Microsoft_Windows

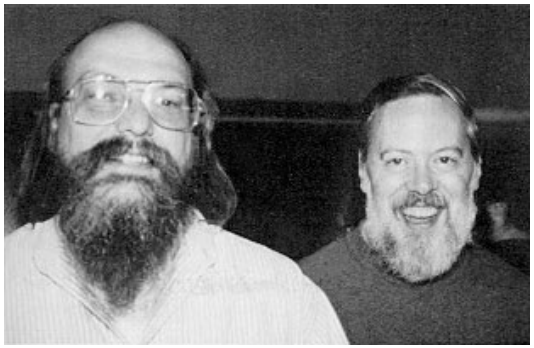
Historia systemu operacyjnego Unix



Wikipedia: History of Unix

Historia Unix

- 1969 Ken Thompson (Bell Labs, AT&T) tworzy na PDP-7 własny system operacyjny, wykorzystując doświadczenia z porzuconego MULTICS
- początek lat 70.: Dennis Ritchie tworzy język C; w 1973 r. Unix zostaje w nim przepisany
- 1974 Thompson i Ritchie publikują opis Uniksa, co przyczynia się do rozpowszechnienia systemu
- od 1977 r. Bill Joy i zespół z Uniwersytetu Kalifornijskiego w Berkeley rozwijają BSD, dodając m.in. obsługę TCP/IP





Historia Unix

- 1983 firma AT&T komercjalizuje Unix System V
- brak standaryzacji różnych wersji Unix-a
  - IBM – AIX (SVR3)
  - DEC – Ultrix (4.2 BSD)
  - SUN – Solaris (SVR4)
  - HP – HP-UX (SVR4)
  - SGI – IRIX (BSD)
  - Microsoft i Santa Cruz Operations – Xenix (Unix Version 7)
- 1988 POSIX, Portable Operating System Interface,
  standard definiujący interfejsy systemowe i programowania aplikacji (API) na poziomie użytkownika, w celu zapewnienia zgodności oprogramowania (przenośności) z wariantami Uniksa i innych systemów operacyjnych

Historia GNU/Linux

- 1984 Richard Stallman ogłasza projekt GNU (*GNU's Not Unix*), którego celem jest stworzenie wolnego systemu operacyjnego zgodnego z Uniksem
- 1985 powstają Free Software Foundation i *The GNU Manifesto*
- 1989 ukazuje się pierwsza wersja licencji GNU GPL
- 1991 Linus Torvalds publikuje pierwszą wersję jądra Linux; w 1992 r. jądro zostaje objęte licencją GNU GPL
- 2007 powstaje The Linux Foundation





comp.os.minix: wpis LBT z 25.08.1991

```
Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and
professional like gnu) for 386(486) AT clones. This has been
brewing since april, and is starting to get ready. I'd like any
feedback on things people like/dislike in minix, as my OS resembles
it somewhat (same physical layout of the file-system (due to
practical reasons) among other things).

I've currently ported bash(1.08) and gcc(1.40), and things seem to
work. This implies that I'll get something practical within a few
months, and I'd like to know what features most people would want.
Any suggestions are welcome, but I won't promise I'll implement
them :-)

Linus (torvalds@kruuna.helsinki.fi)

PS. Yes - it's free of any minix code, and it has a multi-threaded
fs. It is NOT portable (uses 386 task switching etc), and it
probably never will support anything other than AT hard disks, as
that's all I have :-(.
```

GNU/Linux

- GNU (*GNU's Not Unix*) to projekt wolnego systemu operacyjnego zgodnego z Uniksem
- GNU/Linux to system złożony z programów GNU i jądra Linux; projekt GNU rozwija także własne jądro, GNU Hurd
- GNU dostarcza m.in. kompilator GCC, bibliotekę glibc, powłokę Bash i pakiet GNU Coreutils (programy GNU https://www.gnu.org/software/)
- GNU General Public License http://www.gnu.org/licenses/gpl.html
  Powszechna Licencja Publiczna GNU http://www.gnu.org/licenses/gpl.html
  Copyleft w licencjach GNU odwróciła tradycyjne prawo autorskie w celu zapewnienia wolności użytkownikom oprogramowania
- dystrybucje Linuksa: Slackware (1993), Debian (1993), Ubuntu (2004), Red Hat, ...

Rozwój GNU/Linux

Data	Wersja	Linie kodu	
1991/09	0.01	10 K (1K assemblera)	88 plików
1991/11	0.11		
1992/02	0.12		GNU GPL
1992/03	0.95		X Window
1994/04	1.0.0	176 K	
1995/03	1.2.0	311 K	arch. Alpha, SPARC i MIPS, szyna PCI
1996/06	2.0.0	470 K	procesory SMP, SMB
1999/01	2.2.0	1.8 M	
1999/12	2.2.13		
2001/01	2.4.0	3.4 M	ISA Plug-And-Play, USB, PCMCIA, arch. 64 bit, LVM
2003/12	2.6.0	5.9 M	NuMA, hiperwątkowość
2011/07	3.0	14.6 M	
2012/09	3.2	15.9 M	
2014/12	3.18	19.0 M	
2015/07	4.0	22.2 M	
2017/08	4.13	24.8 M	
2020/01	4.19	27.8 M	
2020/08	5.8	28 442 673	70K plików
2024/10	6.11.3		
2026/08	7.0	ponad 40M	wersja stabilna

https://kernel.org/

- 1998 Eric S. Raymond i Bruce Perens zakładają Open Source Initiative (OSI), promując określenie *open source* obok terminu *free software*.
- Na początku lat 2000. firmy Oracle, Informix, Sybase, IBM, Dell i HP deklarują wsparcie dla ruchu open source.
- 2000: SUSE Linux Enterprise Server obsługuje m.in. architekturę IBM S/390 i x86.
- 2001: IBM zapowiada inwestycje w rozwój Linuksa o wartości 1 mld USD i angażuje w ten obszar swoich programistów.
  Microsoft CEO S. Ballmer: *„Linux is a cancer [...] intellectual property destroyer"*
- 2002 Red Hat Linux Advanced Server – system klasy enterprise wspierany przez firmy Dell, IBM, HP, Oracle
- 2007 projekt Samba otrzymuje (po procesie) od Microsoftu dokumentację SMB
- 2009 Microsoft wnosi do jądra Linuksa ponad 20 tys. linii kodu związanych z obsługą Hyper-V
- 2014 Satya Nadella (CEO firmy Microsoft): *Microsoft loves Linux*
- 2016: Microsoft udostępnia Windows Subsystem for Linux i dołącza do Linux Foundation jako członek platynowy.

Rozwój jądra Linux (raport z 2020 r.)

- jeden z największych projektów programistycznych rozwijanych przez społeczność
- regularne wydania co 8-10 tygodni, każde z istotnymi zmianami i usprawnieniami
- w raporcie z 2020 r. szacowano około 12 tys. zmian na wydanie i około 10,7 commitów na godzinę
- 1600 programistów z ponad 200 korporacji
- około 12% zmian pochodziło od niezależnych programistów
- około 52% zmian pochodziło od programistów firm, m.in. Intel, Red Hat, IBM, SUSE, Linaro, Google, Samsung, AMD, Renesas, Texas Instruments i Oracle
- łącznie od 2005 r. ponad 15 600 programistów z 1400 firm wniosło wkład w rozwój

Współczesny rynek systemów operacyjnych

Komputery osobiste (PC, laptopy)

- Windows 73%
- OS X 15%
- Linux 4%
- Chrome OS 2%

Superkomputery

- Linux (TOP 500) 100%

Serwery (2021)

- Linux/Unix 77%
- Windows Server 22.7%

Tablety

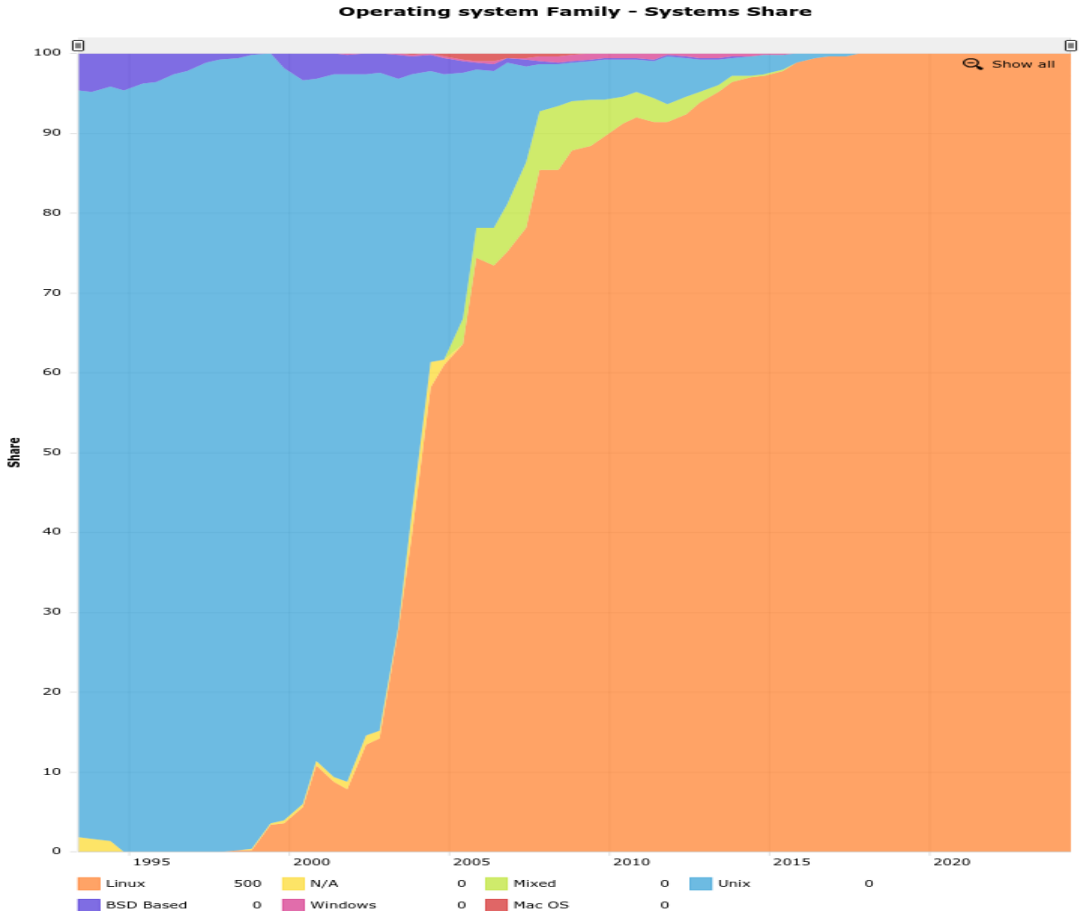
- iOS 55%
- Android 44%

Mobilne

- Android 72%
- iOS 27%

https://gs.statcounter.com
Wikipedia Usage share of operating systems

TOP 500: udział systemów operacyjnych (6-2024)



https://www.top500.org/statistics/overtime/

Pytania sprawdzające

1. Jakie są dwa podstawowe zadania systemu operacyjnego?
2. Co to jest jądro systemu operacyjnego?
3. Jakie są różnice między architekturą harwardzką a architekturą von Neumanna?
4. Na egzaminie nie będzie pytań dotyczących historii ale należy znać pojęcia takie jak, np.: spooling, praca wsadowa, monitor, wieloprogramowość, POSIX, ...