

A wide-angle photograph of a snowy, rocky landscape. The sun is high in the sky, creating a strong lens flare and casting long, dark shadows of the rocks and sparse trees onto the snow. The terrain is uneven with many small mounds and depressions in the snow.

# Systemy Operacyjne

Rodzaje i architektury systemów operacyjnych

Dr hab. inż. Krzysztof Rzecki, prof. AGH

Na podstawie: Abraham Silberschatz, *Koncepcje systemów operacyjnych*



# System operacyjny

**System operacyjny** - zarządza sprzętem komputerowym, jest pośrednikiem między użytkownikiem, a sprzętem komputerowym

**Sprzęt komputerowy** - CPU, RAM, I/O devices

Zadania systemu operacyjnego:

- dostarczenie środowiska do uruchamiania i zarządzania (ang. *control program*) programami użytkownika (**wygoda**),
- dystrybucja zasobów (ang. *resource allocator*) do efektywnej eksploatacji sprzętu komputerowego (**wydajność**).

Użytkownicy	
Programy użytkowe	
System operacyjny	Powłoka
	Jądro i moduły
BIOS - <i>Basic Input Output System</i>	
Sprzęt komputerowy	



# System operacyjny jako program sterujący

Program sterujący (ang. *control program*):

- Nadzorowanie działania programów użytkowych.
- Przechwytywanie i przeciwdziałanie błędom.
- Udostępnianie systemu komputerowego użytkownikom.
- Kontrola dostępu użytkowników i programów do zasobów.
- Obsługa i kontrola pracy urządzeń wejścia-wyjścia.

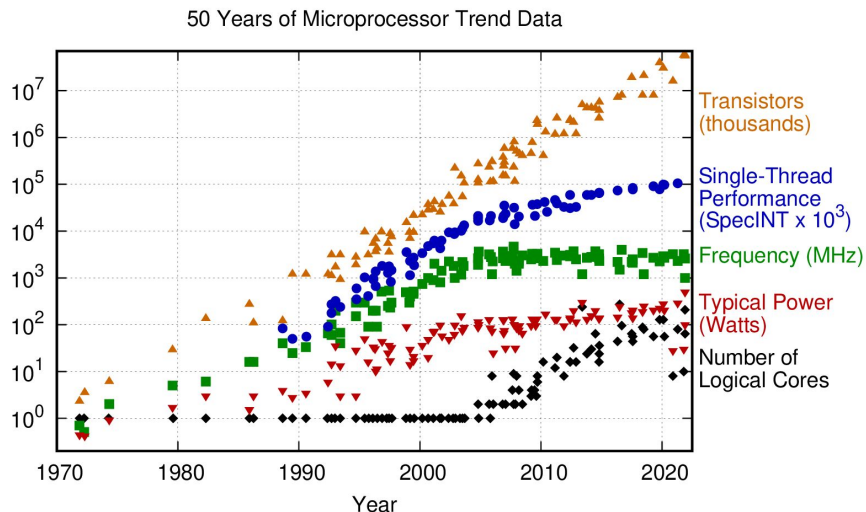


# Dystrybucja zasobów

Dystrybucja zasobów obejmuje:

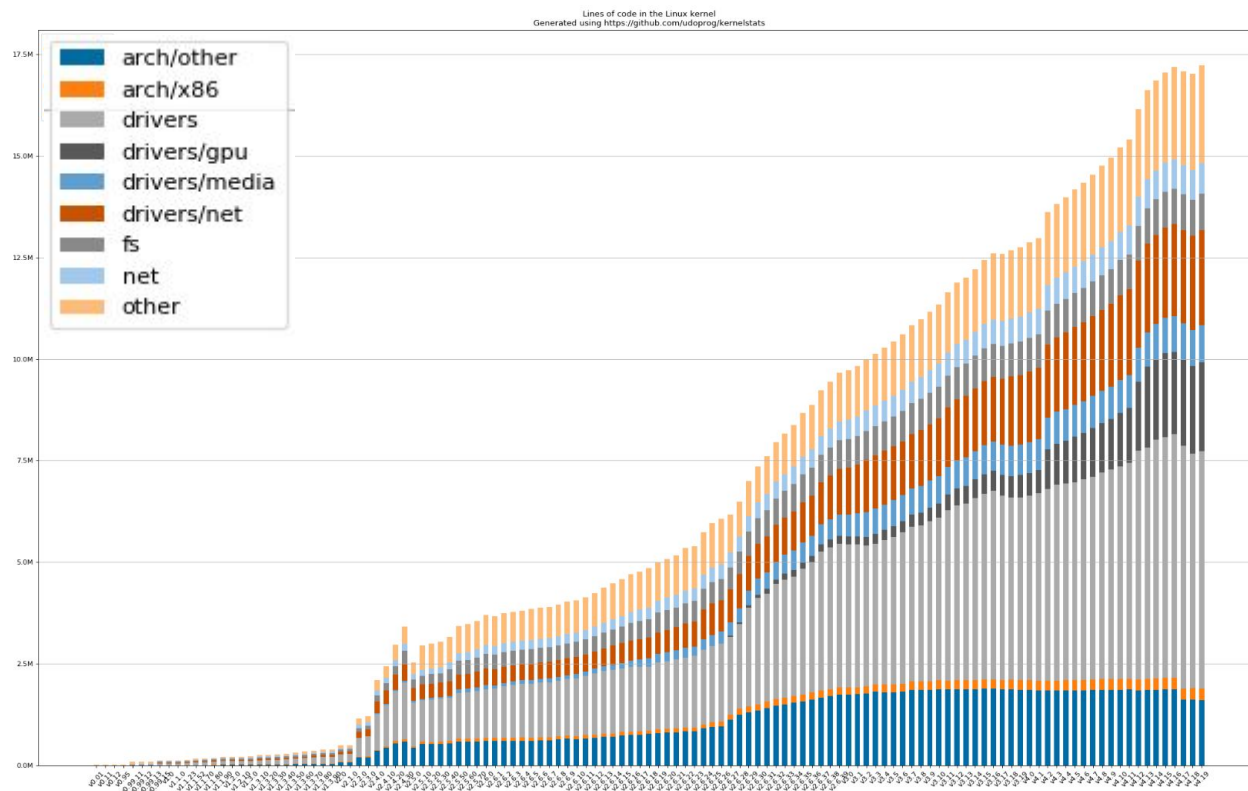
- Planowanie i przydział czasu procesora.
- Kontrola i przydział pamięci operacyjnej.
- Zarządzanie pozostałymi zasobami, jak oprogramowanie czy dostęp do sieci internet.
- Dostarczenie mechanizmów do synchronizacji zadań i komunikacji między zadaniami.

# Rozwój sprzętu - prawo Moore'a



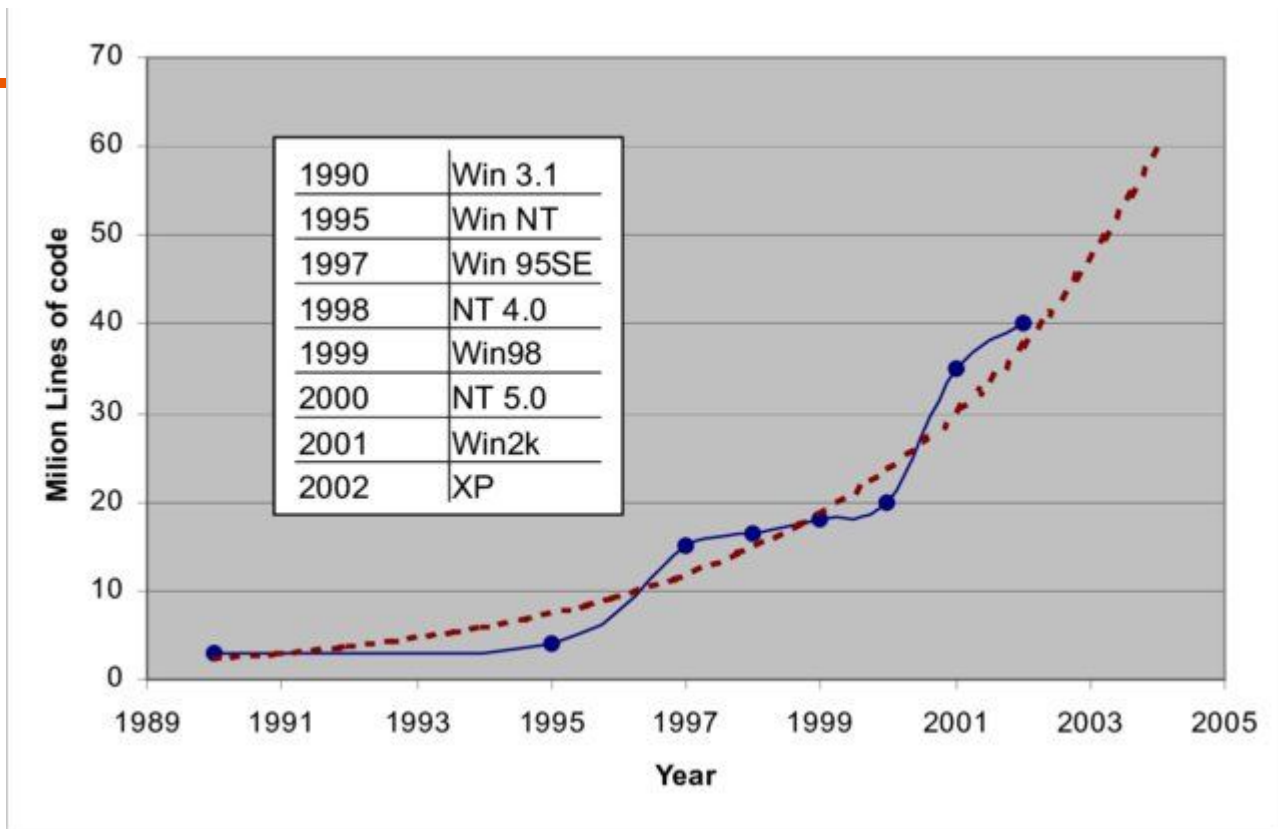
Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten  
New plot and data collected for 2010-2021 by K. Rupp

Źródło: <https://github.com/karlrupp/microprocessor-trend-data>



Źródło: <https://github.com/udoprogram/kernelstats>





Źródło: <https://github.com/udoprogram/kernelstats>



# Podział systemów komputerowych

- Ze względu na wielkość:
  - Mainframe
  - Minicomputer
- Ze względu na zasoby:
  - Server
  - Workstation





# Czy system operacyjny to program ?

- System operacyjny to jedyny program działający cały czas na komputerze
  - (uwaga: systemy wbudowane)
- Program ten zwykle nazywamy: **jądro** (ang. *kernel*)
- Pozostałe typy programów:
  - Programy systemowe (część z nich to polecenia systemowe)
  - Programy aplikacyjne (aplikacje użytkowe)



# Jednostka danych

- **bit** - podstawowa jednostka informacji, oznaczenie: b
- bit może przyjmować wartość 0 lub 1
- kombinacja bitów reprezentuje cyfry, litery, obrazy, wideo, dźwięki, dokumenty, programy, itp.
- **bajt** to 8 bitów, oznaczenie B
- **word** to natywna dla danej architektury komputera jednostka informacji
- np. w architekturze 64-bitowej słowo to 8 bajtów
- Mnożniki:
  - $1 \text{ KB} = 1024 \text{ B} = 2^{10} \text{ B}$
  - $1 \text{ MB} = 1024 \text{ KB} = 1024^2 \text{ B} = 2^{20} \text{ B}$  (małe k =  $10^3$ , wielkie K =  $2^{10}$ )
  - $1 \text{ GB} = 2^{30} \text{ B}$ , itd.
  - $1 \text{ Kib} = 1024 \text{ b}$  i analogicznie  $\text{Mib} = 2^{20} \text{ b}$ ,  $\text{Gib} = 2^{30} \text{ b}$ , itd.

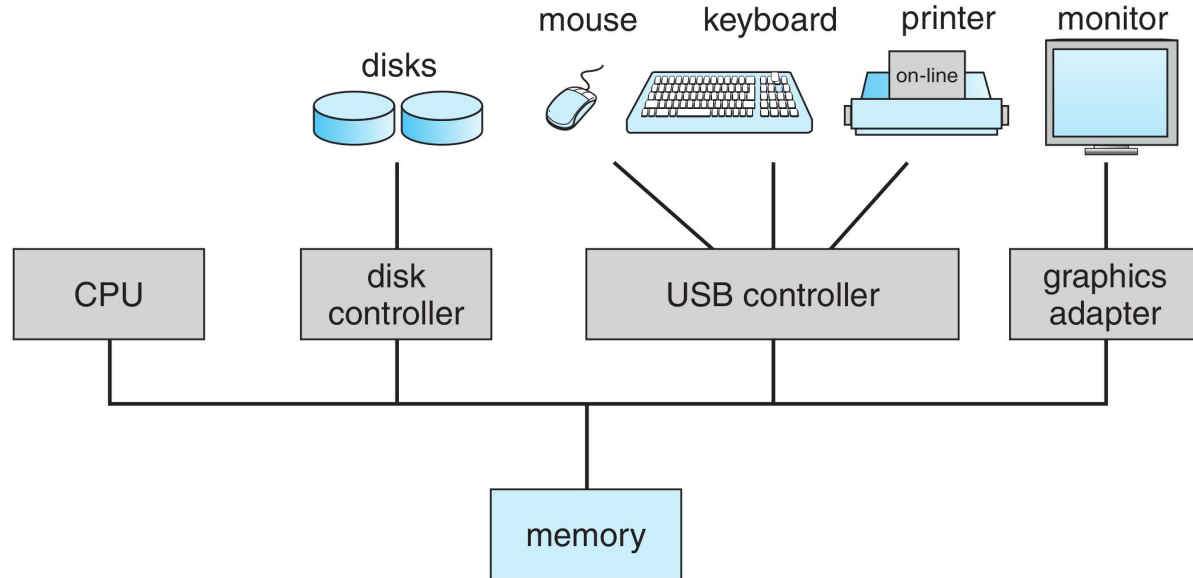
*Młody informatyk myśli, że 1 kilobajt to 1000 bajtów, stary informatyk jest przekonany, że 1 kilometr to 1024 metry...*



# Elementy systemu komputerowego

- Sprzęt: procesor CPU, pamięć RAM, urządzenia we-wy
- Systemy operacyjne: Linux/Unix, Windows, MacOS, itp.
- Programy użytkowe: aplikacje, systemy baz danych, gry komputerowe, oprogramowanie biurowe, środowiska programistyczne, itp.
- Użytkownicy: ludzie, programy, maszyny

# Organizacja systemu komputerowego





# Działanie systemu komputerowego

- Uruchamianie systemu komputerowego:
  - BIOS - Basic Input-Output System
  - Bootstrap program przechowywany w ROM - read-only memory lub EEPROM - erasable programmable read-only memory zwany zwykle firmware
  - Następuje znalezienie i załadowanie systemu operacyjnego oraz pierwszego procesu "init"
  - Oczekiwanie na zdarzenia
- Wystąpienie zdarzenia sygnalizowane jest przez przerwanie (ang. *interrupt*):
  - Sprzętowe przerwanie może być wyzwolone przez przesłanie sygnału przez szynę systemową do procesora
  - Programowe przerwanie może być wyzwolone specjalną operacją, tzw. *system call* lub *monitor call*
- W momencie wystąpienia przerwania, procesor przerywa aktualnie wykonywaną operację, wykonuje procedurę przewidzianą dla danego zdarzenia i wraca do przerwanej operacji.



# Struktura pamięci

- RAM - *random access memory* - pamięć, do której ładowane są programy do wykonania oraz dane dla tych programów
- Pamięci RAM to zwykle technologia półprzewodnikowa DRAM - *dynamic random-access memory*
- ROM - *read-only memory* - pamięć, w której przechowywane są niezmiennalne programy, np. wgrane gry do konsol video
- EEPROM - *erasable programmable read-only memory* - pamięć, która nie może być zbyt często nadpisywana, np. system operacyjny urządzeń mobilnych
- Rejestr - szybka, niewielka pamięć przy/w procesorze na czas wykonywania pojedynczej instrukcji
- Pamięć stała: dysk twardy HDD - *hard disk drive*, dysk optyczny, pen drive, itp.
- Pamięć NVRAM - *nonvolatile RAM* - odpowiedni pamięci RAM podtrzymywany bateryjnie



# Proste systemy wsadowe

- Pierwsze komputery:
  - Wejście: czytniki kart i przewijaki taśm
  - Wyjście: drukarki wierszowe, przewijaki taśm, perforatory kart
- Zadanie na karcie perforowanej: program, dane, karty sterujące
- Czas obliczeń: minuty+ (czasem dni)
- System operacyjny umieszczony na stałe w pamięci operacyjnej
- Grupowanie zadań o podobnych wymaganiach: wsad (ang. *batch*)
- Komputer obsługiwał operator, który pobierał i sortował programy
- Istotna różnica w szybkości działania procesora w porównaniu z we/wy
- Następstwem było wprowadzenie technologii dyskowej





# Wieloprogramowe systemy wsadowe

- Zastosowanie pamięci o dostępie swobodnym (dysków)
- Wczytywanie kart na dysk i zapamiętanie położenia danych => spooling
- Pula zadań (ang. *job pool*) - wczytanie pewnej liczby zadań na dysk
- Możliwość dobierania zadań z dysku tak, aby zwiększyć efektywność jednostki centralnej
- Planowanie zadań (ang. *scheduling*) - planowanie zadań i planowanie przydziału procesora



# Systemy z podziałem czasu (1960r.)

- Problemy systemów wieloprogramowych:
  - wielowariantowość ścieżek wykonywania
  - brak możliwości modyfikowania programu
  - długi czas od rozpoczęcia tworzenia programu do wyniku jego działania
- Podział czasu, wielozadaniowość, ang. *multitasking*
- Interakcyjność, ang. *hands-on* - wymiana danych z programem w ciągu jego trwania
- System plików (ang. *file, filesystem*):
  - zestaw powiązanych informacji
  - format, typ
  - organizacja w katalogi
- Bezpośredni dostęp użytkownika do komputera (bez operatora)
- Pamięć wirtualna - wspomaganie pamięci operacyjnej pamięcią dyskową
- Problem zakleszczenia - wzajemne oczekiwanie programów na zasoby



# Systemy dla komputerów osobistych ('70)

- Zmniejszenie cen sprzętu
- Rozwój linii komputerów PC (ang. *personal computer*)
- Początkowo systemy operacyjne dla pierwszych komputerów osobistych: nie wielostanowiskowe, nie wielozadaniowe, lecz z czasem je rozwinięto
- Microsoft: MS-DOS, później Microsoft Windows
- IBM: MS-DOS (od Microsoft), później OS/2
- Apple: Macintosh, później iOS
- Bell Laboratories: UNIX dla PDP-11 (wiele koncepcji z systemu MULTICS dla komputera GE645)
- Na bazie rozwiązań UNIX w latach '80 => Windows NT, IBM OS/2, Macintosh Operating System



# Systemy wieloprocessorowe, równoległe

- Zwiększona przepustowość
- Współczynnik przyspieszenia, który nie zawsze powiela wydajność
- Współużytkowanie urządzeń zewnętrznych
- Zwiększenie niezawodności (redundancja/nadmiarowość węzłów obliczeniowych)
- Wieloprzetwarzanie symetryczne - w każdym procesorze działa identyczna kopia
  - Działa N procesów na N egzemplarzach jednostki centralnej
  - Może się zdarzyć niebalansowanie obciążenia procesorów
  - Wersja Encore systemu UNIX dla komputera Multimax
  - SunOS w wersji 5 (Solaris 2) dla komputerów Sun
- Wieloprzetwarzanie asymetryczne - każdy procesor ma inne zadanie (+procesor główny)
  - Np. słabsze procesory obsługują komunikację (ang. *front-end*)
  - IBM i komputer IBM Series/1 jako procesor czołowy
  - SunOS w wersji 4 dla komputerów Sun

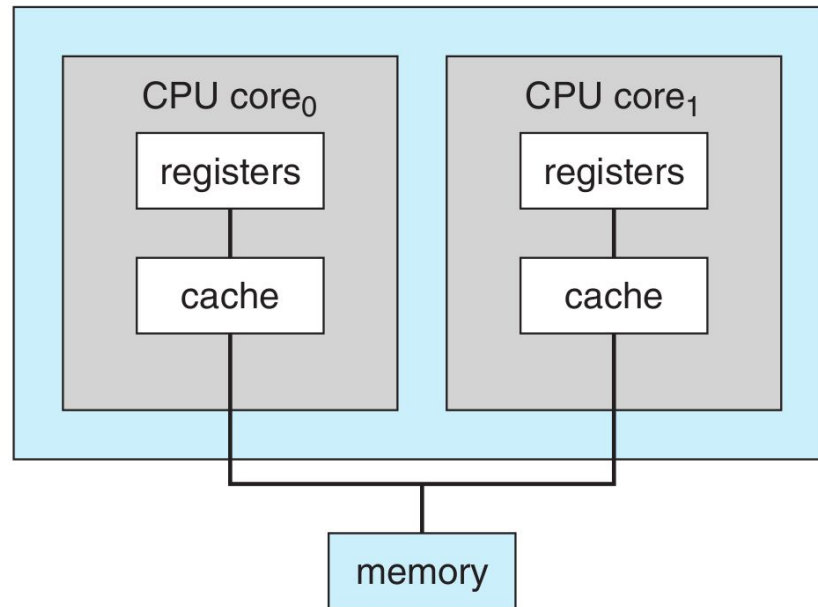


# UMA/NUMA

- UMA - *uniform memory access* - pamięć o jednorodnym czasie dostępu
- NUMA - *non-uniform memory access* - pamięć o niejednorodnym czasie dostępu

System operacyjny musi tak zarządzać dostępem do pamięci, aby minimalizować efekt NUMA.

# Procesor wielordzeniowy





# Systemy rozproszone, systemy klastrowe

- Systemy luźno powiązane (ang. *loosely coupled*), rozproszone (ang. *distributed systems*)
- Rozdzielenie geograficzne
- Połączenie przez linie telekomunikacyjne (internet, linie telefoniczne, itp.)
- Zróżnicowanie architektur poszczególnych węzłów (ang. *node*)
- Zróżnicowanie mocy obliczeniowych poszczególnych węzłów
- Cechy:
  - Podział zasobów
  - Przyspieszenie obliczeń
  - Niezawodność
  - Komunikacja



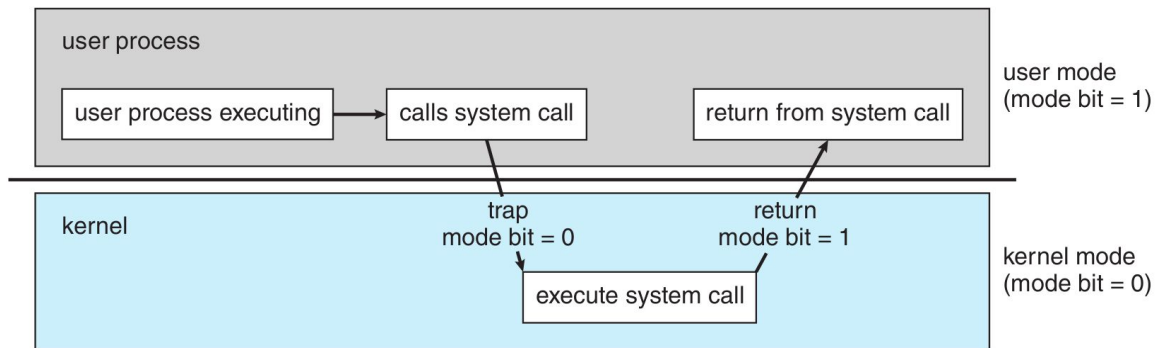


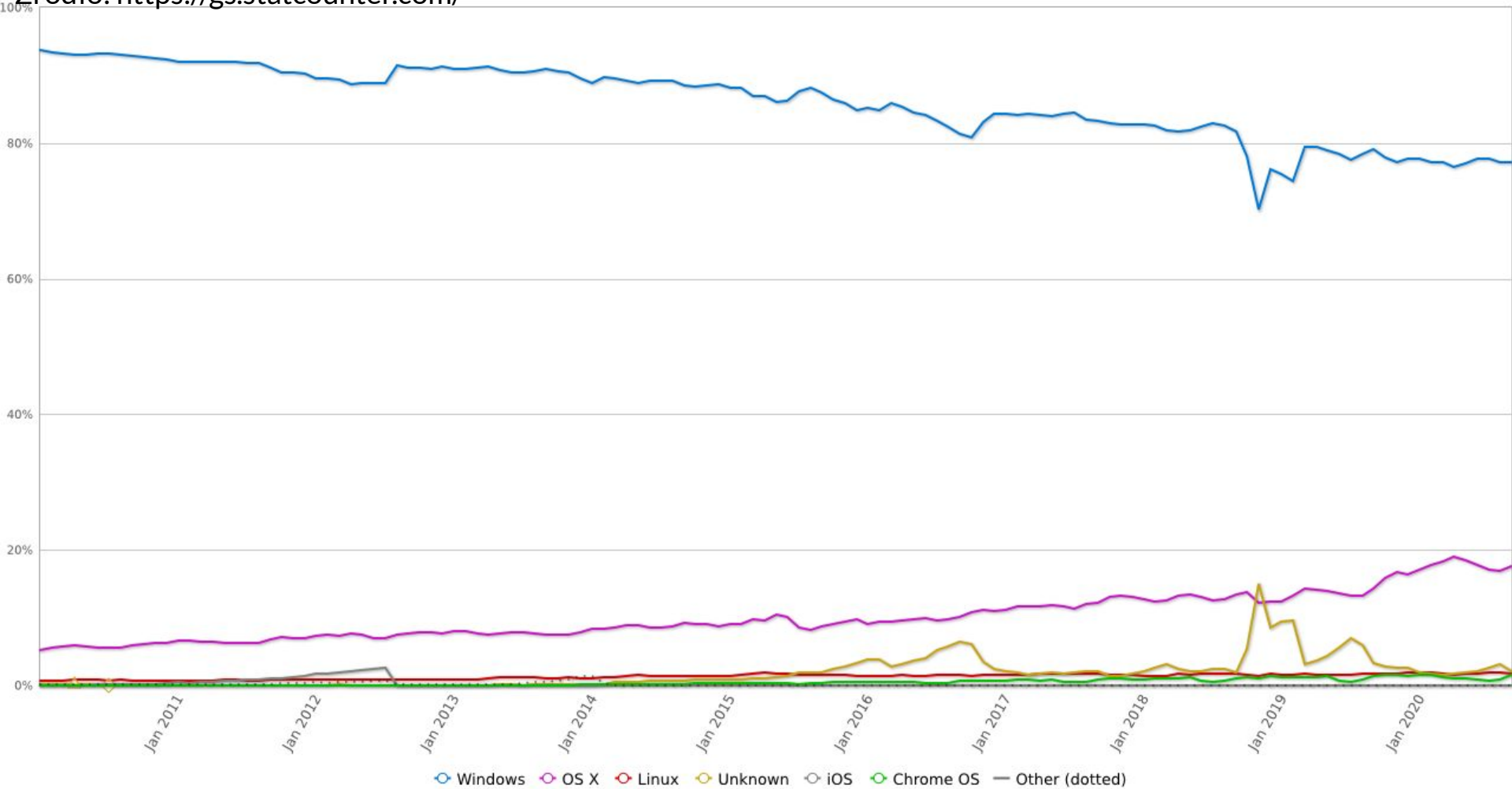
# Systemy czasu rzeczywistego (ang. *real-time*)

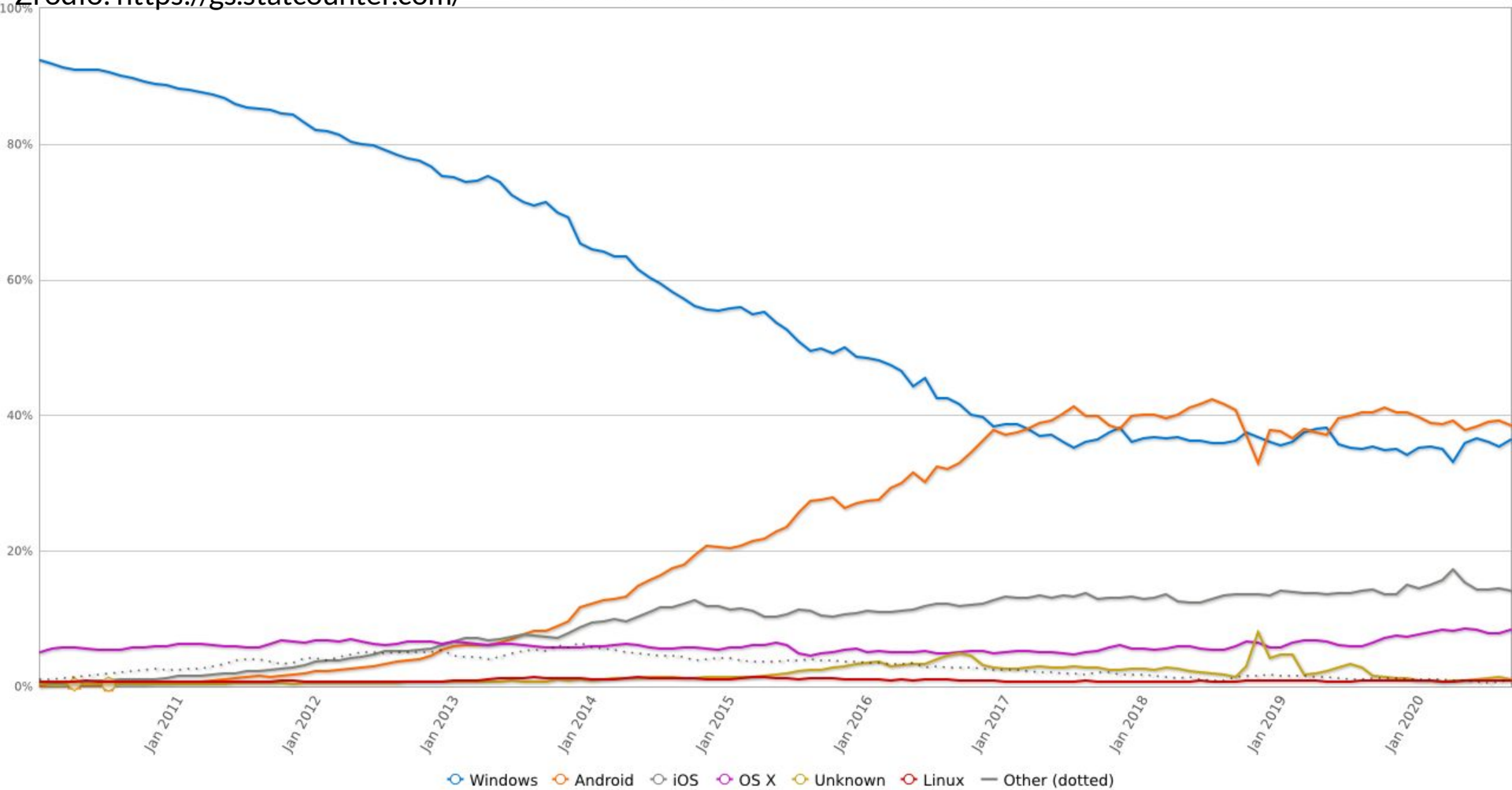
- Zastosowanie: surowe wymagania na czas wykonania operacji lub przepływu danych
- Przykłady:
  - Jednokierunkowe sterowanie maszyną według zadanego programu
  - Odczytywanie wartości czujników
  - Analiza odczytanych wartości czujników i adekwatna reakcja robota
  - Analiza otoczenia, przetwarzanie danych (sygnałów, obrazów) i podejmowanie decyzji
- Odmiany systemów czasu rzeczywistego:
  - Rygorystyczny (ang. *Hard real-time system*) - terminowe wypełnienie krytycznych zadań
  - Łagodny (ang. *Soft real-time system*) - krytyczne zadanie do obsługi otrzymuje pierwszeństwo

# Tryby pracy systemu operacyjnego

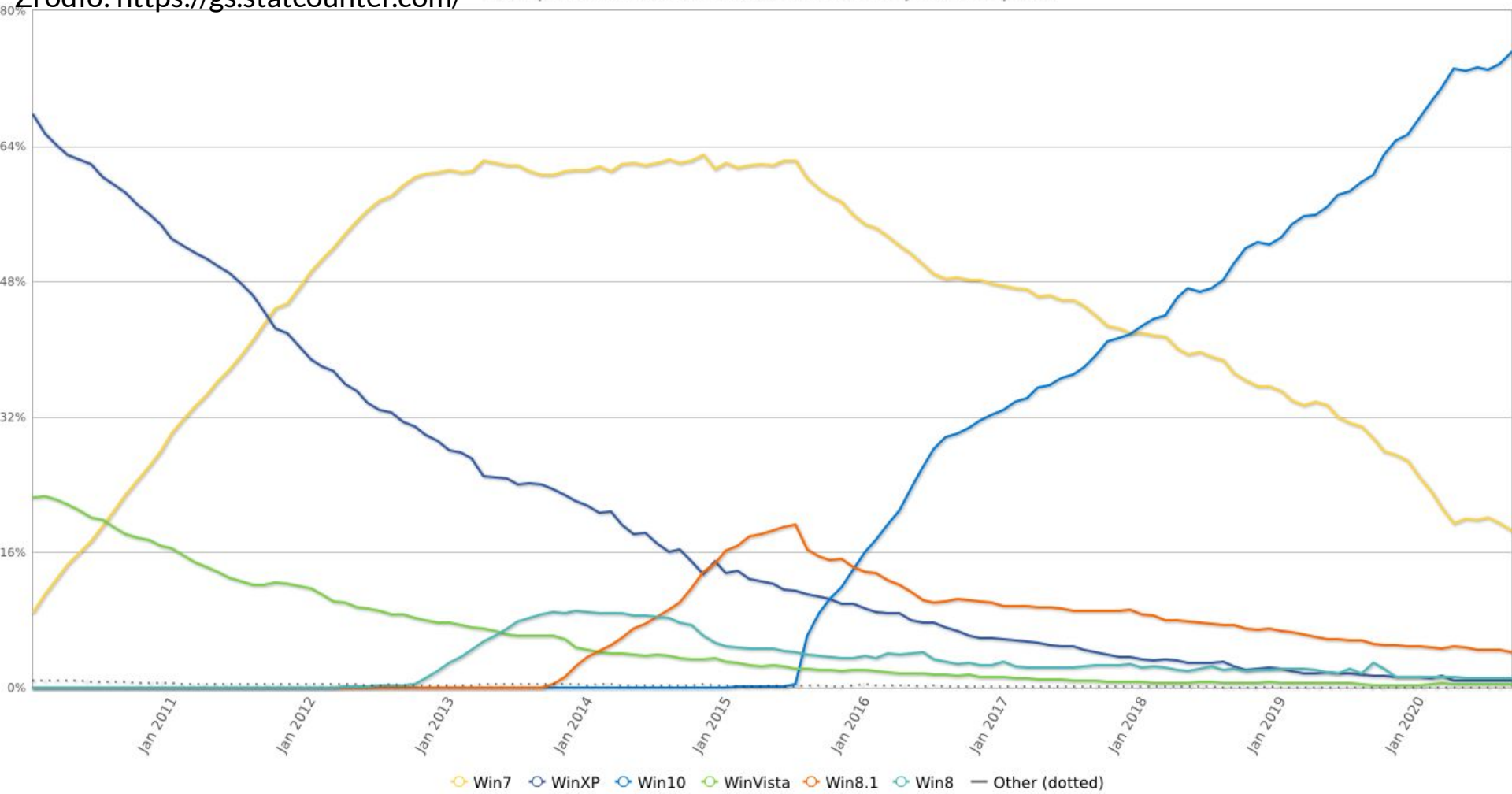
- Tryb użytkownika (ang. *user mode*)
- Tryb jądra (ang. *kernel mode*)
- Zaimplementowany w sprzęcie *mode bit*: 0 - kernel mode, 1 - user mode

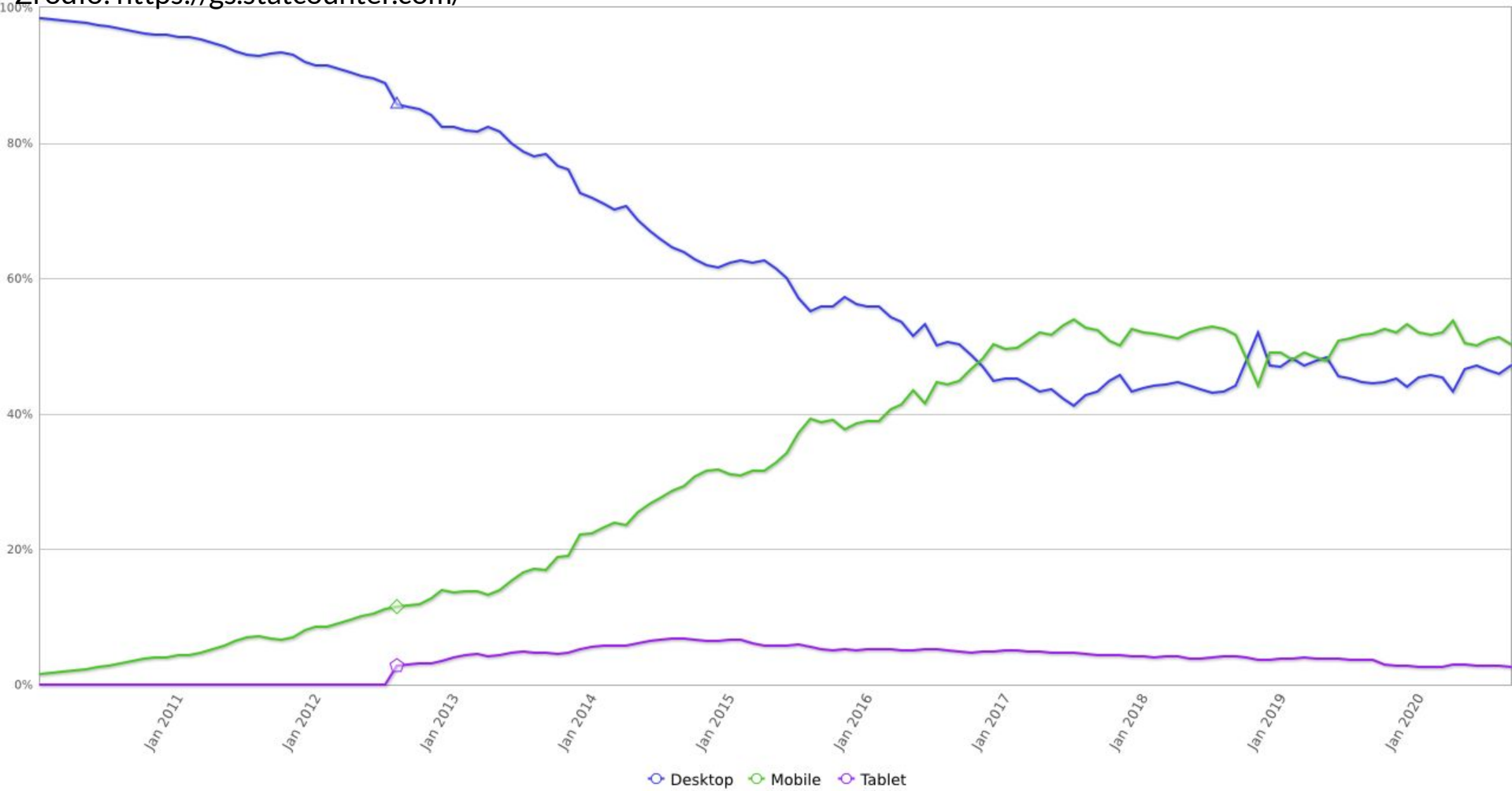






**StatCounter Global Stats**  
Desktop Windows Version Market Share Worldwide from Jan 2010 - Sept 2020





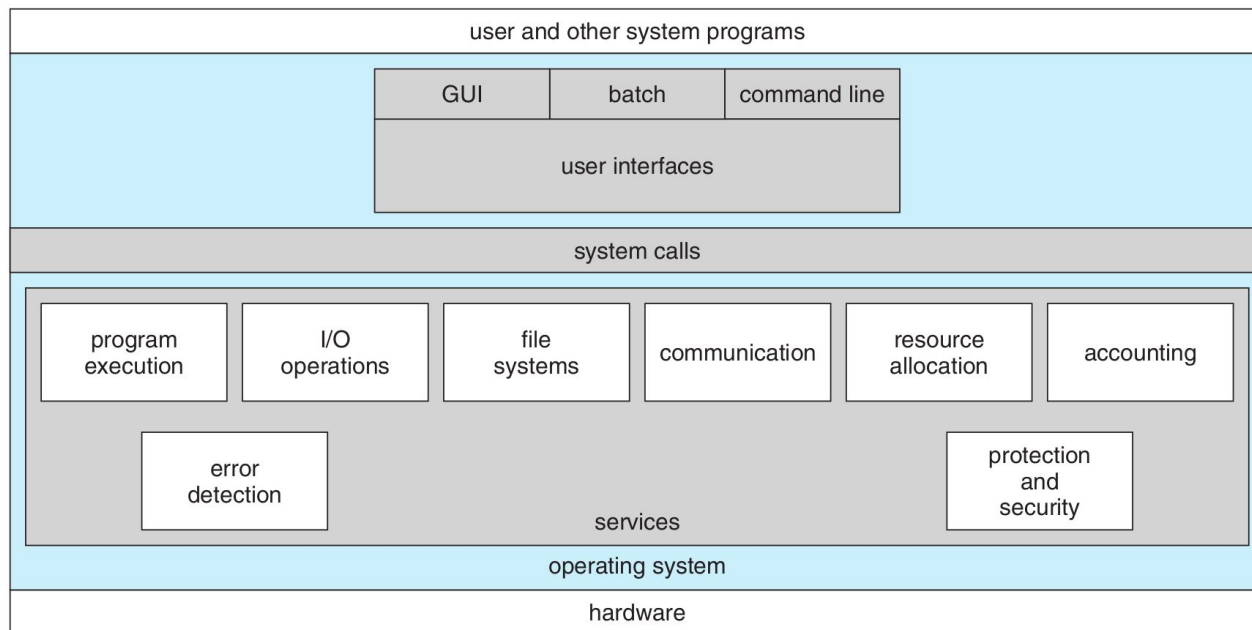


# Pytania...

- Jakie zagrożenia niesie za sobą wielodostęp ?
- Co jest główną zaletą wieloprogramowości ?
- Czy współcześnie może mieć zastosowanie system bez dysku twardego ?
- Jakie uzasadnienie istnienia mają systemy rozproszone ?
- Jaka jest największa trudność w tworzeniu systemu czasu rzeczywistego ?
- Jakie znaczenie ma tryb jądra i tryb użytkownika dla bezpieczeństwa systemu ?



# Struktura systemu operacyjnego





# Interfejs użytkownika

- Graficzny interfejs użytkownika (GUI - graphical user interface) to najpopularniejszy wśród użytkowników biurowych interfejs człowiek-komputer.
- Interpreter poleceń (CLI - command-line interface) jest interfejsem użytkownika (UI) przyjmującym komendy tekstowe oraz funkcje.
- Interfejs *batch* to zbiór komend i dyrektyw kontrolujących te komendy zapisanych w pliku, który po uruchomieniu kolejno uruchamia komendy z tego pliku.
- Interpreter poleceń == *shell*: sh (Bourn shell), bash (Bourne-Again shell), csh (C shell), zsh (Z shell), ksh (Korn shell), itp.
- Interpretacja poleceń: polecenie systemowe (np. cd) lub program (np. rm).



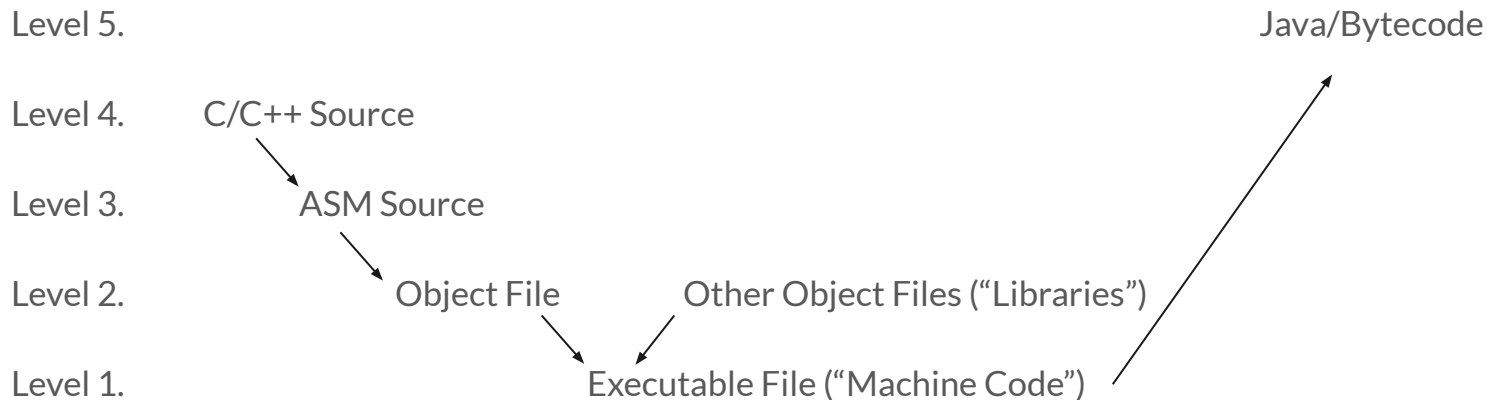
# Podstawowe polecenia bash

- Dokumentacja na temat poleceń dostępna jest (wychodzimy klawiszem 'q'):

`$ man <nazwa polecenia lub programu>`

- **Polecenia:** pwd, cd, ls, cat, cp, mv, mkdir, rm, touch, locate, find, grep, df, du, head, tail, diff, tar, chmod, chown, jobs, kill, ping, wget, history, man, echo, zip, unzip, hostname, useradd, userdel, curl, df, diff, echo, exit, finger, free, grep, groups, less, passwd, ping, shutdown, ssh, reboot, sudo, top, uname, w, whoami
- **Instrukcje:** instrukcjami skryptowymi: if, for, while, until, etc.
- **Edytory tekstu:** vim, pico, etc.

# Poziomy języków programowania





**Skrzyczne**