Systemy operacyjne

Tomasz Lewicki

WWSIS, Wrocław

marzec 2007

Czym jest system operacyjny?

Mianem **systemu operacyjnego** określa się program (zbiór programów) pośredniczących w komunikacji między użytkownikiem a sprzętem komputerowym.

W dalszej części wykładu określenie *system operacyjny* będzie często zastępowane skrótem SO. W zależności od kontekstu zamiennie będą używane określenia *system komputerowy* i *urządzenie komputerowe* (urządzenie). W odniesieniu do urządzeń wejścia i wyjścia pojawi się skrót *urządzenia we/wy*.

Zadania systemu operacyjnego

- umożliwienie wygodnego korzystania z urządzenia
- ułatwienie użytkownikowi wykonania pewnego zadania bądź grupy zadań wykonywanych jednorazowo lub cyklicznie
- efektywny podział i maksymalizacja wykorzystania zasobów udostępnianych przez system komputerowy
- kontrolowanie działania urządzenia komputerowego

System komputerowy

Na system komputerowy składa się kilka elementów, które można przedstawić w układzie warstwowym. Warstwy wyższe nie mogą funkcjonować bez warstw leżących niżej.

użytkownik

↑

programy użytkownika

↑

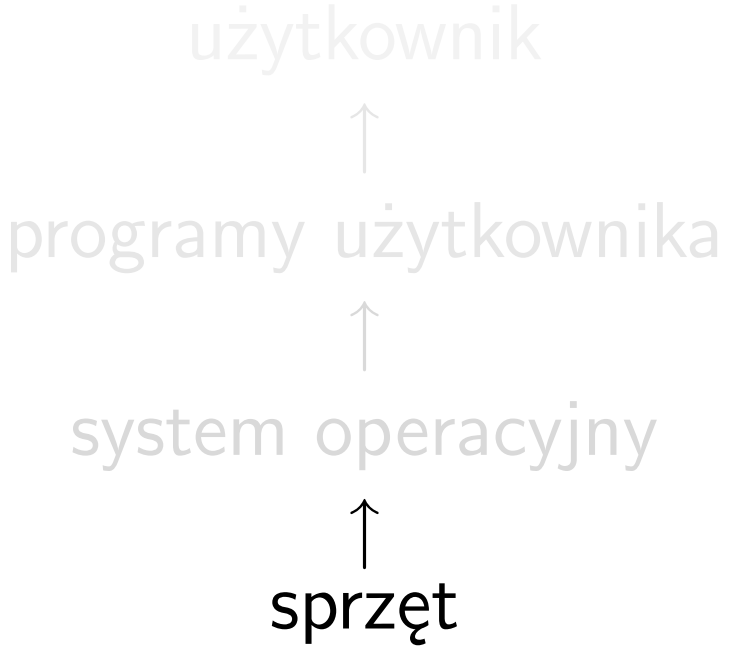
system operacyjny

↑

sprzęt

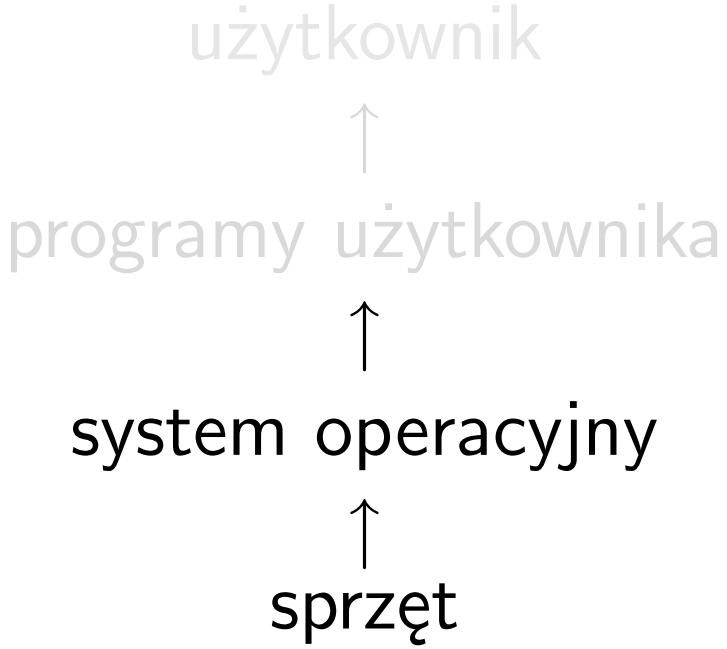
System komputerowy

Na system komputerowy składa się kilka elementów, które można przedstawić w układzie warstwowym. Warstwy wyższe nie mogą funkcjonować bez warstw leżących niżej.



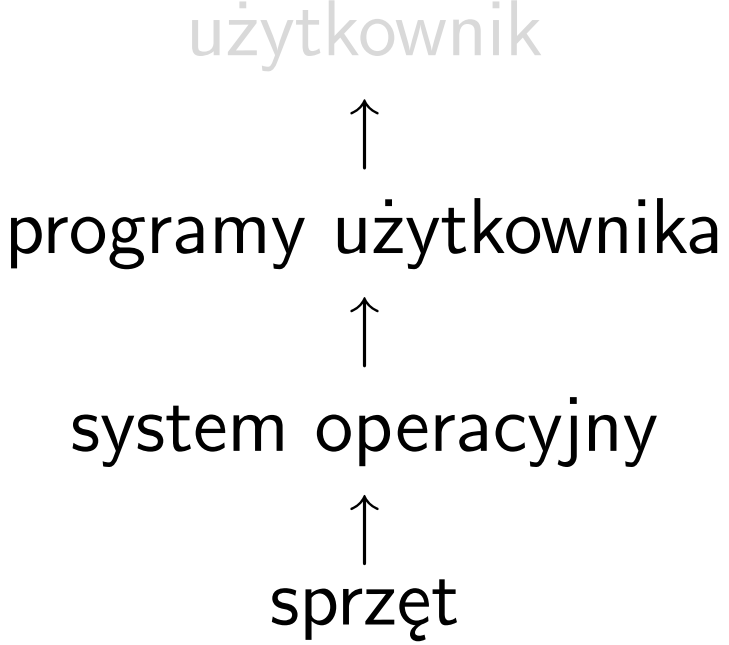
System komputerowy

Na system komputerowy składa się kilka elementów, które można przedstawić w układzie warstwowym. Warstwy wyższe nie mogą funkcjonować bez warstw leżących niżej.



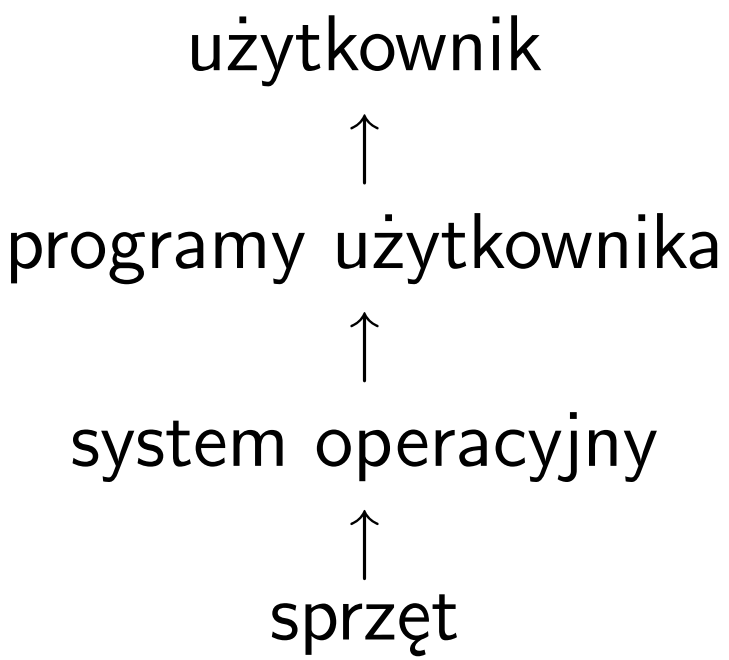
System komputerowy

Na system komputerowy składa się kilka elementów, które można przedstawić w układzie warstwowym. Warstwy wyższe nie mogą funkcjonować bez warstw leżących niżej.



System komputerowy

Na system komputerowy składa się kilka elementów, które można przedstawić w układzie warstwowym. Warstwy wyższe nie mogą funkcjonować bez warstw leżących niżej.



System komputerowy — c.d.

Poszczególne warstwy systemu komputerowego, poczynając od najniższej, to:

- **sprzęt** (*hardware*) — zapewnia podstawowe zasoby, m.in. procesor (CPU), pamięć główną i pomocniczą, urządzenia **we**jścia/**wy**jścia (*Input/Output*, I/O)
- **system operacyjny** — pośredniczy między użytkownikami i sprzętem, nadzoruje wykonywanie zadań i przydziela zasoby
- **programy użytkownika** (aplikacje, *software*) — służą do wykonywania określonych zadań (edycja tekstów, bazy danych, kompilacja programów, edukacja, zarządzanie projektami, . . .)
- **użytkownik** — może nim być zarówno człowiek, jak i inny komputer lub urządzenie komputerowe

Maszyna wirtualna

Mianem **maszyny wirtualnej** określa się sprzęt pracujący pod kontrolą systemu operacyjnego i łatwiejszy w obsłudze dla użytkownika.

Inaczej: maszyna wirtualna to „surowy” sprzęt w postaci użytecznej dla końcowego użytkownika, wymagającej od niego niewielkiej wiedzy co do wewnętrznej budowy sprzętu i interakcji między jego elementami.

Maszyna wirtualna — c.d.

Przykłady zastosowania koncepcji maszyny wirtualnej

- pamięć masowa i system plików — adresowanie plików i katalogów za pomocą wskaźników fizycznych (np. ścieżka (cylinder)/sektor/blok w przypadku dysku twardego) jest zastąpione tzw. *ścieżkami dostępu*
- urządzenia we/wy — odpowiednie sterowniki dbają np. o prawidłowe wyświetlanie obrazu, interpretację znaków wprowadzanych z klawiatury czy wydruk wyników na urządzeniu drukującym
- ochrona zasobów i obsługa błędów — ochrona zarówno przed przypadkowymi naruszeniami bezpieczeństwa i spójności danych, jak i złośliwym działaniem ze strony innych użytkowników
- sterowanie programami — użytkownik może wykonywać potrzebne mu operacje za pomocą różnych interfejsów, np. graficznego (*GUI*) czy linii poleceń (*shell*)

Wymagane cechy systemu operacyjnego

- planowanie i szeregowanie zadań oraz zarządzanie kolejnością ich wykonania
- sterowanie przerwaniami
- obsługa wyjątków i błędów
- planowanie przydziału zasobów i ich ochrona
- realizacja zasady *wielodostępu* do zasobów
- obsługa urządzeń we/wy

Pożądane cechy systemu operacyjnego

- wysoka wydajność (kryteria: wykorzystanie procesora i innych zasobów sprzętowych, średni czas wykonania zadania, czas reakcji na polecenie, czas przełączania między zadaniami)
- duża niezawodność
- mały rozmiar
- łatwość utrzymania, aktualizacji i ewentualnej rozbudowy
- przyjazność dla użytkownika

Nieco historii

Systemy operacyjne są nierozerwalnie związane z historią informatyki. Pierwsze systemy powstały w latach 50. XX wieku i *de facto* nie były „prawdziwymi" SO. Były to najczęściej pojedyncze programy stworzone w ściśle określonym celu i wykonujące tylko jedno zadanie, np. obliczenia balistyczne (amerykańska maszyna *ENIAC*) albo łamanie szyfrów (brytyjska maszyna *Colossus*).

Kolejnym, coraz doskonalszym modelom maszyn liczących przypisywano kolejne zadania, które mogły być wykonywane zamiennie, w zależności od aktualnych potrzeb wyrażanych przez użytkowników. Tak narodziły się „prawdziwe" systemy operacyjne, a wiele koncepcji zastosowanych w pierwszych systemach klasy *mainframe* przetrwało do dziś i możemy się z nimi spotkać nawet w systemach przenośnych.

System jednoużytkownikowy

- jeden program sterujący (brak „prawdziwego” SO)
- programista był jednocześnie operatorem systemu
- sterowanie maszyną za pomocą przełączników
- nośniki danych i wyników w postaci taśm papierowych i kart perforowanych

Największą wadą pierwszych systemów operacyjnych było nieefektywne wykorzystanie czasu procesora, przez co obliczenia na nich wykonywane były drogie (same systemy i ich utrzymanie też sporo kosztowały). Wiele czasu pochłaniały też typowo techniczne czynności: zakładanie i zmiany taśm lub kart z danymi oraz wynikami, programowanie maszyny (zmiana pozycji przełączników) oraz konserwacja.

Prosty system wsadowy

- użytkownicy o różnych wymaganiach przygotowywali własne zestawy zadań i danych dla maszyny liczącej oraz przekazywali je operatorowi
- operator mógł przeanalizować dostarczone przez użytkowników zadania i pogrupować je według podobnych wymagań (np. języka programowania)
- zadania o podobnych wymaganiach mogły być wykonywane jedno po drugim, ich rozpoznawanie odbywało się za pomocą kart sterujących

Systemy wsadowe znacząco skróciły czas wykonywania zadań na maszynach liczących. Poprawiło się również wykorzystanie czasu procesora. Do wad można zaliczyć długi czas przepływu zadania od użytkownika do operatora i z powrotem oraz „wąskie gardło" w postaci powolnych podajników kart i/lub przewijaków taśm oraz drukarek wyników obliczeń.

Złożony system wsadowy

Spool oraz multiprogramming

- zadania o podobnych wymaganiach rezydują w pamięci operacyjnej
- jeśli jedno z zadań czeka na zakończenie długotrwałej operacji (np. trwają obliczenia), można wykonać inną operację dla kolejnego zadania, np. odczytać kolejną porcję danych lub wydrukować wyniki
- *spool* (**s**imultaneous **p**eripherial **o**peration **o**n–**l**ine) — podczas wykonywania obliczeń pewnego zadania jednocześnie wykonywane są operacje we/wy dla innych zadań
- *multiprogramming* — w pamięci operacyjnej znajduje się wiele zadań, procesor zajmuje się kolejno każdym z nich

Złożone systemy wsadowe przypominają współczesne wielozadaniowe systemy operacyjne. Pojawiają się w nich pojęcia *planowania przydziału procesora*, *szeregowania zadań*, *zarządzania pamięcią*, *ochrony zadań*, *przydziału urządzeń*.

System wielozadaniowy (z podziałem czasu)

- w pamięci operacyjnej jednocześnie znajduje się wiele zadań
- zadania nie mieszczące się w pamięci operacyjnej są przenoszone do pamięci wirtualnej (we współczesnych systemach na dysk twardy), następuje wymiana danych między pamięcią operacyjną i wirtualną (ten proces nazywamy *swappingiem*)
- procesor jest kolejno przydzielany poszczególnym zadaniom
- przełączanie między zadaniami odbywa się bardzo szybko — użytkownik ma wrażenie jednoczesnej pracy z wieloma programami (wielozadaniowość, *multitasking*)
- na jednej maszynie może pracować jednocześnie wielu użytkowników

Wszystkie nowoczesne SO charakteryzują się wielozadaniowością i podziałem czasu procesora.

Inny podział systemów operacyjnych

Systemy operacyjne można również podzielić ze względu na obszar zastosowania i — pośrednio — rodzaj sprzętu, na jakim działają.

- systemy biurkowe (*desktop*)
- systemy równoległe
- systemy rozproszone
- systemy klastrowe
- systemy czasu rzeczywistego
- systemy wbudowane i przenośne

Systemy biurkowe

Przeznaczone dla komputerów osobistych (*personal computer*, PC) używanych do zastosowań domowych i biurowych. Pierwsze urządzenia tego typu pojawiły się w latach 70. XX w. Początkowo przeznaczone dla jednego użytkownika, z czasem wyewoluowały do rozbudowanych systemów zdolnych obsłużyć wielu użytkowników jednocześnie (cecha przejęta z komputerów typu *mainframe*). Położono w nich nacisk na wygodę użytkowania, interaktywność i duży wybór urządzeń we/wy.

Wraz z rozwojem mikroelektroniki, upowszechnieniem coraz większych i szybszych pamięci (operacyjnych oraz masowych), szybszych procesorów i urządzeń we/wy systemy biurkowe zaczęły rozwijać się bardzo burzliwie, z maszyn 8-bitowych ewoluując na 16-, 32- i 64-bitowe. Prawdziwą rewolucję wywołało jednak upowszechnienie Internetu.

Znane systemy biurkowe to rodzina Microsoft Windows, rozmaite dystrybucje Linuksa, rodzina BSD czy MacOS X firmy Apple.

Systemy równoległe

Przetwarzanie symetryczne i asymetryczne

Przeznaczone dla komputerów wieloprocesorowych dzielących wspólną szynę systemową, zegar, pamięć i urządzenia we/wy. Systemy te określamy również mianem *ściśle związanych* (*tightly coupled*). Zalety takich systemów to większa wydajność, wyższa niezawodność i większa odporność na awarie i błędy.

Systemy równoległe dzielą się na **symetryczne** (*Symmetrical Multiprocessing*, SMP) oraz **asymetryczne** (*Asymmetrical Multiprocessing*, AMP). W systemach symetrycznych każdy procesor posiada własną kopię SO, procesy są wykonywane jednocześnie, a procesory komunikują się za pośrednictwem wspólnej szyny. Większość współczesnych SO pozwala na uruchomienie w trybie SMP. W systemach asymetrycznych główny procesor (*master*) przydziela zadania procesorom podrzędnym (*slave*). To drugie rozwiązanie jest spotykane raczej w bardzo dużych systemach.

Systemy równoległe — c.d.

Wieloprocesorowe przetwarzanie równoległe i NUMA

Konkurencją dla rozwiązania SMP jest **wieloprocesorowe przetwarzanie równoległe** (*Multiprocessor Parallel Processing*, MPP), stosowane w niektórych superkomputerach. W tej technologii procesory mają oddzielne pamięci operacyjne i szyny danych.

Technologia **niejednolitego dostępu do pamięci** (*Non-Uniform Memory Access*, NUMA), czasem nazywana CC-NUMA (*Cache Coherent NUMA*) zastępuje SMP w rozbudowanych serwerach. Koncepcja polega na podziale całej struktury serwera na węzły zawierające kilka procesorów (zazwyczaj od czterech do ośmiu). Każdy węzeł dysponuje częścią ogólnej pamięci serwera, ale ma dostęp do całej pamięci. Dodatkowo każdy węzeł posiada własną pamięć podręczną. Dostęp do pamięci podręcznej (*cache*) i lokalnej pamięci węzła (*local node memory*) jest znacznie szybszy i rzadszy niż odwołania do pamięci innych węzłów, przez co szyna danych jest mniej obciążona i może obsłużyć większą liczbę procesorów.

Systemy rozproszone

Są odmianą systemów równoległych z pamięcią lokalną oddzielną dla każdego procesora. Systemy komunikują się poprzez różnego rodzaju media, np. sieci lokalne i rozległe, sieci telefoniczne czy szybkie szyny danych, dlatego też nazywa się je niekiedy systemami *luźno związanymi* (*loosely coupled*).

Zaletami systemów rozproszonych są:

- podział zasobów
- szybsze wykonywanie obliczeń
- rozkładanie obciążeń
- wyższa niezawodność
- możliwość komunikacji między węzłami

Systemy rozproszone są jednymi z najszybciej rozwijających się SO. Przykłady: sieci P2P, rozwiązania klient–serwer.

Systemy klastrowe

Systemy klastrowe (klastry, *clusters*) różnią się od systemów równoległych tym, że są oddzielnymi, niezależnymi systemami operacyjnymi dzielącymi pamięć masową i łączą się za pośrednictwem sieci lokalnej. Ich główną zaletą jest niezawodność; są używane w zastosowaniach, w których wymagana jest ciągłość dostępu do zasobów (tzw. *high availibility service*). Dodatkowo są wydajniejsze niż pojedyncze serwery.

Klastry o tzw. *wysokiej dostępności* dzielą się na **symetryczne** (*symmetric clustering*), zwane też **aktywny–aktywny** i **asymetryczne** (*asymetric clustering*), zwane również **aktywny–pasywny**. W pierwszym przypadku wszystkie węzły klastra działają jednocześnie i monitorują się nawzajem, w drugim — jeden z węzłów jest zapasowy, włącza się w przypadku awarii innego węzła.

Inne rodzaje klastrów: klastry w sieciach rozległych (np. sieci *Storage Area Networks* (SAN)), klastry równoległe.

Systemy czasu rzeczywistego

Systemy czasu rzeczywistego (*real–time systems*, RTS) stosowane są w urządzeniach, w których czas jest najważniejszym parametrem sterującym; nierzadko posiadają sprzężenie zwrotne.

Rozróżnia się dwa rodzaje RTS: **rygorystyczne** („twarde", *hard RTS*) i **łagodne** („miękkie", *soft RTS*). Pierwszy typ wykorzystywany jest m.in. do sterowania procesami przemysłowymi, w obrazowaniu medycznym, do nadzorowania eksperymentów naukowych, w niektórych systemach sprzedaży, w bibliotekach. Drugi typ spotyka się we współczesnych systemach operacyjnych; wymagania czasowe są złagodzone, dopuszczalne są opóźnienia w wykonywaniu innych zadań. Typowe zastosowania *soft RTS* to multimedia, gdzie zadaniu wymagającemu działania w czasie rzeczywistym przydziela się wysoki priorytet.

Systemy wbudowane i przenośne

Systemy wbudowane (osadzone, *embedded*) i przenośne (*handheld*) to specyficzny rodzaj systemów operacyjnych, charakteryzujący się małymi rozmiarami, ograniczoną liczbą funkcji i niewielką możliwością rozbudowy. Wymiary urządzeń i stopień upakowania elementów elektronicznych w obudowie narzucają ograniczenie na ilość pamięci, prędkość procesora, wielkość ekranu i jakość wyświetlanego przez niego obrazu, ilość i rodzaje dostępnych portów rozszerzeń oraz inne właściwości. Do tych surowych wymogów musi zostać dostosowany system operacyjny i działające pod jego kontrolą aplikacje użytkowe.

Rola urządzeń przenośnych i wbudowanych w nie systemów operacyjnych szybko rośnie wraz ze wzrastającymi wymaganiami użytkowników.

Przykłady urządzeń, w których można spotkać systemy wbudowane i przenośne: *Personal Digital Assistant* (PDA), Palm PC, Pocket PC, telefony komórkowe, samochodowe komputery pokładowe.

Architektura systemów operacyjnych

Istnieją trzy koncepcje budowy systemów operacyjnych:

- jednolita (monolityczna) — zbiór procedur wywołujących się wzajemnie bez ograniczeń (przykład: Linux)
- warstwowa — procedury zgrupowane są w moduły, a te z kolei w warstwy. Poszczególne moduły są zależne tylko od warstw leżących niżej, przez co od warstw najniższych wymaga się najwyższej niezawodności (przykład: Solaris)
- klient–serwer — procedury zgrupowane są w moduły traktowane mniej lub bardziej równorzędnie. Dwukierunkowa komunikacja między modułami nie odbywa się bezpośrednio; odpowiada za nią specjalny program, zwany **mikrojądrem** (przykład: Windows NT)

Składniki systemu operacyjnego

- zarządzanie procesami
- zarządzanie pamięcią operacyjną
- zarządzanie pamięcią podręczną
- zarządzanie pamięcią masową
- zarządzanie plikami
- zarządzanie urządzeniami we/wy
- kontrola błędów i obsługa wyjątków
- mechanizmy kontroli dostępu do zasobów
- usługi użytkownika
- usługi sieciowe

Terminologia

- procesor, proces, program, wątek, przetwarzanie równoległe (współbieżne), szeregowanie zadań, planiści
- komunikacja międzyprocesowa, synchronizacja, przerwanie, pułapka, semafor, blokada
- pamięć operacyjna (główna), pamięć masowa (pomocnicza), pamięć podręczna (*cache*), strony (płaty, bloki) pamięci, zarządzanie pamięcią
- przestrzeń adresowa, adresowanie
- dzielenie i ochrona zasobów, tryb nadzorcy, tryb użytkownika
- systemy plików
- sterowniki urządzeń, szyna komunikacyjna, szyna systemowa, bufor
- interfejs, interfejs sieciowy, interfejs użytkownika
- wielozadaniowość, wielodostęp
- wywłaszczanie, wymiatanie, stronicowanie