



Funkcje systemu operacyjnego

Użytkownik komputera osobistego klasy PC (oprócz sprzętu) musi posiadać specjalne oprogramowanie, które pozwala na interakcję pomiędzy człowiekiem i maszyną — **system operacyjny** (ang. *Operating System*, OS).

Historia systemów operacyjnych sięga lat 50. XX w., kiedy zaczęto wykorzystywać pierwsze elektroniczne maszyny liczące do celów militarnych. Początkowo nie były to OS-y z prawdziwego zdarzenia, a jedynie pojedyncze programy. Z czasem dołączano kolejne podprogramy i funkcje, które ostatecznie przybrały formę znanego współcześnie systemu operacyjnego.

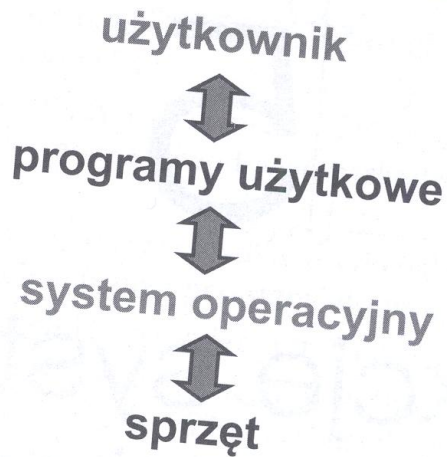
Największą popularność (w komputerach osobistych klasy PC) zyskały komercyjne systemy spod znaku Microsoftu (MS-DOS, Windows 3.x, Windows (9x) czy rozwijana do dzisiaj seria Windows NT — NT, 2000, XP, Vista, 7, 8, 8.1, 10). Istnieją jednak alternatywne, nie mniej popularne rozwiązania, np. rozpowszechniany na licencji **GNU GPL Linux** (duża rodzina niezależnych dystrybucji), **FreeBSD**, **Open Solaris** czy komercyjne **Uniksy**: **Oracle Solaris** (wcześniej **Sun Solaris**), **Mac OS X** itd.

Niezależnie od tego, który system operacyjny został wybrany przez użytkownika, aby można było z niego skorzystać, należy go wcześniej **zaimplementować na określonej platformie sprzętowej** — czyli po prostu **zainstalować**.

5.1. Pojęcie systemu operacyjnego

System operacyjny to oprogramowanie, które stanowi interfejs pomiędzy **użytkownikiem**, **oprogramowaniem użytkowym** i **urządzeniami komputera osobistego** — te składniki **systemu komputerowego** przedstawia rysunek 5.1.

Dzięki OS użytkownik może **sterować i zarządzać sprzętową platformą komputerową**, wydając polecenia w sposób **bezpośredni** (wpisuje polecenia w wierszu — interfejs znakowy) lub **pośredni** (korzysta z **interfejsu graficznego**).



Rysunek 5.1. Układ warstwowy składający się na system komputerowy

Główne zadania systemu operacyjnego to: zarządzanie procesami, pamięcią operacyjną, pamięcią podręczną cache i pamięciami masowymi, implementowanie systemu plików i zarządzanie nim, zarządzanie urządzeniami I/O, kontrola błędów i obsługa wyjątków, mechanizmy kontroli dostępu do zasobów i uwierzytelniania użytkownika, zarządzanie usługami sieciowymi.

Dobry system operacyjny powinien:

- mieć interfejs przyjazny dla użytkownika;
- cechować się wysoką wydajnością i stabilnością;
- mieć stosunkowo niewielki rozmiar;
- oferować możliwość aktualizowania i modyfikowania;
- udostępniać dużą liczbę aplikacji;
- posiadać sterowniki do popularnych urządzeń.

5.2. Rodzaje systemów operacyjnych

Wraz z rozwojem platform sprzętowych ewoluowały systemy operacyjne — dzisiaj można wyodrębnić kilka klasyfikacji OS na podstawie różnych kryteriów podziału.

5.2.1. Podział ze względu na liczbę wykonywanych zadań

- **Systemy jednoprogramowe.** Są w stanie wykonywać tylko jedno zadanie zlecone przez użytkownika, przetworzenie kolejnego programu następuje po zakończeniu poprzedniego (np. MS-DOS).
- **Systemy wieloprogramowe.** Umożliwiają wykonywanie kilku programów rezydujących w pamięci dzięki współdzieleniu mocy obliczeniowej mikroprocesora (ang. *multiprogramming*).

- **Systemy wielozadaniowe.** Umożliwiają wykonywanie kilku programów dzięki współdzieleniu mocy obliczeniowej mikroprocesora. Przełączanie pomiędzy poszczególnymi programami (procesami) jest planowane i następuje tak szybko, że użytkownik ma wrażenie ciągłego działania kilku aplikacji (ang. *multitasking*). W tego typu systemach wprowadzono pojęcie **pamięci wirtualnej**, przez co tzw. pamięć logiczna mogła być większa od fizycznej (Windows, Linux).

5.2.2. Podział ze względu na sposób przetwarzania

- **Systemy przetwarzania bezpośredniego.** System interpretuje zadania użytkownika na bieżąco i realizuje je zaraz po uruchomieniu. Istnieje bezpośrednia interakcja pomiędzy interfejsem a użytkownikiem (Windows, Linux).
- **Systemy przetwarzania pośredniego.** Pomiędzy zleceniem zadania przez użytkownika a jego realizacją przez system operacyjny (systemy wsadowe) występuje opóźnienie.
 - » **Prosty system wsadowy.** Dla systemu operacyjnego przygotowywano ciąg operacji w postaci tzw. wsadu. Za jego przygotowanie odpowiadał operator, który po konsultacji z użytkownikami przygotowywał ciąg poleceń zapisanych na karcie perforowanej lub taśmie magnetycznej, realizowanych następnie przez OS.
 - » **Złożony system wsadowy.** Przypominał współczesne systemy wielozadaniowe, w których najczęściej wykonywane zadania są przechowywane w pamięci, a podczas obliczeń mogą być wykonywane operacje I/O dla innych zadań (ang. *simultaneous peripheral operation on-line*) lub kolejne zadania zapisane w pamięci.

5.2.3. Podział systemów wieloprocessorowych

- **Systemy wieloprocessorowe symetryczne SMP** (ang. *Symmetric Multiprocessing System*). Są to systemy operacyjne przeznaczone do komputerów wieloprocessorowych (wielordzeniowych) dzielących wspólną magistralę systemową, zegar, pamięć i urządzenia wejścia-wyjścia. Umożliwiają zlecenie realizacji zadań systemowi dwóm lub więcej mikroprocesorom. Zadania są dzielone symetrycznie pomiędzy wszystkie CPU, dzięki czemu wzrasta ogólna wydajność systemu komputerowego (rodzina Windows NT, Linux).
- **Systemy wieloprocessorowe asymetryczne AMP** (ang. *Asymmetric Multiprocessing System*). Systemem zarządza główny procesor, a jednostki CPU mają zlecane inne, niezależne zadania. Przykładem pracy w układzie asymetrycznym mogą być mikroprocesory współpracujące z zewnętrznym koprocessorem, działające niezależnie, taktowane inną częstotliwością zegara itd.

5.2.4. Podział ze względu na liczbę obsługiwanych użytkowników

- **Systemy jednoużytkownikowe.** Umożliwiają użytkowanie na poziomie jednego użytkownika, brak jest mechanizmów uwierzytelniających czy zabezpieczających dostęp do plików i katalogów lub takie zabezpieczenia są nieskuteczne (MS-DOS, Windows 9x).
- **Systemy wieloużytkownikowe.** Umożliwiają użytkowanie OS przez wielu użytkowników jednocześnie i niezależnie. System może przechowywać profile swoich użytkowników w taki sposób, że inni klienci platformy nie mają dostępu do zawartych w nich danych (rodzina Windows NT, Linux).

5.2.5. Podział ze względu na interfejs użytkownika

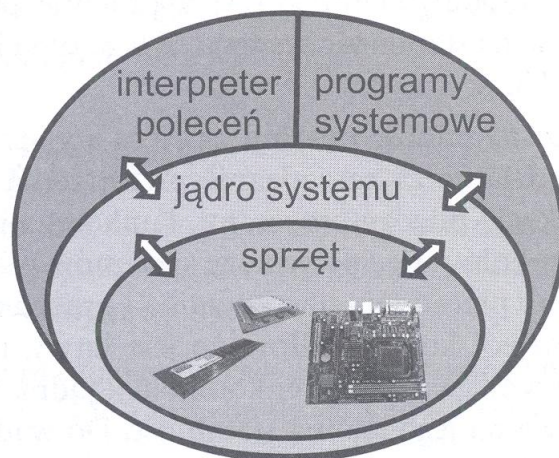
- **Systemy z interfejsem znakowym/tekstowym.** Komunikacja użytkownika z systemem następuje poprzez wprowadzanie tekstowych komend w wierszu poleceń (np. MS-DOS, Linux Bash, Windows Server Core).
- **Systemy z okienkowym interfejsem graficznym.** Możliwe jest użytkowanie systemu z wykorzystaniem interfejsu graficznego opartego na symbolach (ikonach) oraz tzw. okienkach, określanego skrótem GUI (ang. *Graphical User Interface*) (Windows, X Window).

5.2.6. Inne rodzaje systemów operacyjnych

- **Systemy czasu rzeczywistego** (ang. *Real-Time Operating Systems* — RTOS). To systemy operacyjne, których zadaniem jest reagowanie na zmiany zewnętrzne zachodzące w czasie rzeczywistym. Głównym kryterium są ramy czasowe, które warunkują działanie systemu.
Systemy operacyjne czasu rzeczywistego dzielą się na dwa rodzaje:
 - » **Twarde (rygorystyczne).** Znany jest najgorszy (najdłuższy) czas odpowiedzi oraz wiadomo, że nie zostanie przekroczony. Gwarantuje to terminowe wypełnianie krytycznych zadań
 - » **Miękkie (łagodne).** Starają się odpowiedzieć najszybciej, jak to możliwe, ale nie wiadomo, jaki może być najgorszy czas odpowiedzi. Krytyczne zadania mają pierwszeństwo przed innymi, system nie gwarantuje jednak terminowego wypełnienia krytycznych zadań.
- **Systemy rozproszone** (ang. *distributed systems*). Umożliwiają zarządzanie wieloma niezależnymi komputerami połączonymi np. za pomocą wydajnej sieci komputerowej (tzw. klastrem).

5.3. Struktura systemu operacyjnego

Na system operacyjny składają się zazwyczaj: **jądro systemu**, **oprogramowanie systemowe** (m.in. sterowniki) oraz **interpreter poleceń** bądź **GUI** (rysunek 5.2).



Rysunek 5.2. Schemat budowy systemu operacyjnego

Głównym elementem systemu operacyjnego jest jego **jądro** (ang. *kernel*), które stanowi trzon platformy programowej. Jądro jest traktowane jako zbiór procedur, które są odpowiedzialne za bezpośrednie zarządzanie sprzętem komputerowym i udostępniają zestaw usług służących do implementacji oprogramowania systemowego — jest interfejsem pomiędzy sprzętem a oprogramowaniem użytkowym. Można pokusić się o stwierdzenie, że **jądro systemu to właściwie system operacyjny**.

Ważny element systemów operacyjnych stanowi warstwa abstrakcji sprzętu (**HAL** — ang. *Hardware Abstraction Layer*). Jest ogniwem pośredniczącym między sprzętem a jądrem systemu operacyjnego. Warstwa ta jest niezbędna do odseparowania konkretnej architektury systemu komputerowego od oprogramowania użytkowego (dlatego wybrany system operacyjny może działać na komputerach różnych producentów z różną konfiguracją sprzętową). Jest sposobem komunikacji ze sprzętem poprzez udostępnianie odpowiednich funkcji bibliotek i sterowników, dzięki czemu programy (i sam system) komunikują się z urządzeniami na poziomie ogólnym. Systemami korzystającymi z warstwy abstrakcji sprzętowej są np.: Mac OS, Linux, DOS, Solaris, BSD czy Windows z rodziny NT.

Najważniejsze zadania jądra systemowego to: **zarządzanie procesami**, **zarządzanie pamięcią operacyjną**, **obsługa systemu wejścia-wyjścia**, **zarządzanie plikami i przestrzenią dyskową**, **uwierzytelnianie i ochrona** oraz **implementacja interfejsu poleceń**.

Istnieje kilka podstawowych koncepcji budowy jądra systemu operacyjnego:

- **Jądro monolityczne.** Jądro to duży program, którego zadaniem jest wykonywanie wszystkich najważniejszych funkcji i zadań systemu operacyjnego. Zaletą jądra monolitycznego jest szybkość działania (jądro nie jest rozbite na wiele podprogramów). Oprócz tego, jeśli jest ono dobrze przygotowane, może mieć niewielkie rozmiary,

a ponadto w jednym pliku łatwiej wyszukuje się ewentualne błędy. Do wad tego typu jądra można zaliczyć trudności w rozbudowie jednego wielkiego programu, a także to, że błędy związane z jedną częścią oprogramowania mogą wpłynąć na stabilność całego jądra. Źle przygotowane jądro może się także rozrosnąć do dużych rozmiarów, a ponadto moduły i jądro korzystają z jednej przestrzeni adresowej, co również może rzutować na stabilność systemu. Z tego typu jądra korzystają systemy operacyjne z rodziny Unix.

- **Mikrojądro.** Jest ono stosunkowo małe i zawiera wyłącznie mechanizmy niskopoziomowe odpowiedzialne za zarządzanie przestrzenią adresową, zarządzanie wątkami czy komunikację między procesami. Funkcje związane z obsługą sterowników urządzeń, protokołów sieciowych czy systemów plików są przenoszone do specjalnych bloków lub przestrzeni użytkownika i uruchamiane jako moduły. Oto zalety mikrojądra: rozwój kodu źródłowego jest łatwy, implementacja nowych funkcjonalności nie wymaga ponownej kompilacji jądra, błąd w jednym elemencie systemu nie wpływa na jego ogólną stabilność. Do wad zaliczymy dość trudny proces wyszukiwania błędów oraz to, że uruchomienie wielu programów prowadzi do spadku wydajności i większego obciążenia pamięci. Systemem, który korzysta z koncepcji mikrojądra, jest Minix.
- **Jądro hybrydowe.** Jest połączeniem koncepcji dwóch powyższych rozwiązań: łączy szybkość i prostotę konstrukcji jądra monolitycznego z modułowością i bezpieczeństwem mikrojądra. Jądro hybrydowe uruchamia pewne moduły w swojej przestrzeni w celu zmniejszenia utraty wydajności, a jednocześnie przenosi określone funkcjonalności w postaci usług do przestrzeni użytkownika. Dla przykładu: w wewnętrznej strukturze jądra hybrydowego można implementować wirtualny system plików i sterowniki magistrali, natomiast zarządzanie systemem plików i pamięcią masową może zostać przeniesione poza jądro i być uruchamiane jako usługa. Jądra hybrydowe są wykorzystywane w systemach z rodziny Windows NT.

W skład systemu operacyjnego mogą wchodzić również **programy systemowe**, które nie są częścią jądra.

Kolejnym elementem systemu operacyjnego jest **interpreter poleceń**, który może być zaimplementowany w jądrze systemu lub przyjmować postać odrębnego programu, np. interfejsu graficznego.

W zależności od rodzaju zastosowanego jądra system operacyjny może cechować:

- **Wielozadaniowość** (ang. *multitasking*). Umożliwia wykonywanie wielu procesów na zasadzie dzielenia czasu mikroprocesora.
- **Wielodostępność** (ang. *multiuser*). Umożliwia pracę wielu użytkowników systemu operacyjnego w tym samym czasie.
- **Wielowątkowość** (ang. *multithreading*). Umożliwia wykonanie jednego procesu w ramach kilku wątków.
- **Wielobieżność** (ang. *reentrant*). Kilka procesów może mieć dostęp do interfejsu jądra (praca w trybie jądra), dzięki czemu wszystkie mogą korzystać z funkcji systemowych.

- **Skalowalność** (ang. *scalability*). Cecha ta opisuje możliwość łatwej rozbudowy elementów systemu operacyjnego. Ważne jest, aby pomimo zwiększania objętości systemu nie spadała jego wydajność.
- **Wywłaszczanie** (ang. *preemption*). Technika ta pozwala na wstrzymanie jednego procesu, aby możliwe było uruchomienie innego. Zawieszenie jednego procesu nie wstrzymuje całego systemu operacyjnego.

5.4. Systemy plików

Żaden system operacyjny nie może się obejść bez **systemu plików** (ang. *file system*), który zarządza sposobem zabezpieczania i przechowywania plików w przestrzeni pamięci masowej, np. na twardym dysku.

Plik (ang. *file*) jest pewnym ciągiem danych charakteryzującym się skończoną długością oraz pewnymi atrybutami — jest interpretowany przez system operacyjny jako całość.

Większość współczesnych systemów operacyjnych ma własne systemy plików, które ewoluują wraz z rozwojem OS (tabela 5.1). System plików udostępnia użytkownikowi logiczną strukturę (interfejs) w postaci drzewa katalogowego, a sam obsługuje skomplikowane procesy zapisu i odczytu danych. Ponadto system plików zarządza ich dodatkowymi atrybutami (w postaci metadanych) i udostępnia je programom, implementując **mechanizmy kontroli dostępu do plików**. Należy pamiętać, że z poziomu systemu Windows nie odczytamy bezpośrednio systemów plików dla Linuksa (jest to możliwe dopiero po zainstalowaniu odpowiednich programów ze sterownikami pod Windowsa). Natomiast system Linux raczej bez problemu będzie obsługiwał systemy plików przeznaczone dla Windowsa.

Tabela 5.1. Zestawienie kilku popularnych systemów plików

System plików	Symbol	Opis
File Allocation Table 32	FAT32	Wprowadzony przez Microsoft wraz z systemem Windows 95 OSR 2. Wykorzystuje 32-bitową tablicę FAT, co umożliwia zapis pliku o maksymalnej wielkości 4 GiB. Do zaadresowania jednostki alokacji stosuje się 28 bitów, co maksymalnie daje 2^{28} jednostek alokacji. Najmniejsza jednostka alokacji może zawierać 4 kiB, a największa 32 kiB. Ograniczenie rozmiaru woluminu to 2^{32} sektorów, co daje 2 TiB na jednym dysku lub macierzy dyskowej. System nie obsługuje metadanych, co oznacza, że nie implementuje np. zabezpieczenia plików przed nieautoryzowanym dostępem. W przypadku systemu Windows XP i nowszych maksymalna ilość miejsca na dysku z systemem FAT może wynosić do 8 TiB, a wielkość pojedynczego woluminu może wynieść do 32 GiB.

System plików	Symbol	Opis
New Technology File System	NTFS	Został wprowadzony wraz z systemem Windows NT 3.1. Do adresowania wykorzystuje 64 bity, co teoretycznie daje na jeden plik 2^{64} bajtów, czyli 16 EiB; praktycznie jest to 16 TiB. Maksymalny teoretyczny rozmiar partycji wynosi 2^{64} , jednak w rzeczywistości jest to 256 TiB. W przeciwieństwie do FAT obsługuje metadane i w najnowszej odsłonie NTFS 3.1 umożliwia: kompresję, szyfrowanie, nadawanie praw dostępu, transakcyjność oraz księgowanie operacji dyskowych (zalecany dla systemów z rodziny Windows NT).
Resilient File System	ReFS	Następca systemu plików NTFS. Dostępny na razie w systemach Windows Server 2012 i nowszych (docelowo także w wersjach dla użytkowników domowych). Obsługuje duże woluminy (do 1 jo-bibajta — YiB), udostępnia tworzenie pul pamięci masowej, zapewnia łatwą wirtualizację oraz odporność na wszelkiego rodzaju uszkodzenia danych. Pamiętać jednak należy, iż dyski systemowe (boot/OS) nie mogą korzystać z systemu plików ReFS.
Extended File System wersja 4	EXT4	Czwarta wersja popularnego systemu plików przeznaczonego do systemów operacyjnych z rodziny Linux. EXT4 używa adresowania 48-bitowego, umożliwia więc stosowanie woluminów o wielkości do 1 EiB, a maksymalny rozmiar pliku to 16 TiB. Pozwala na przechowywanie do 32 000 podkatalogów, księgowanie operacji, obsługę metadanych, nadawanie praw dostępu, rezerwację miejsca bez fragmentacji dla nowych plików.
B-tree File System	BtrFS	Został zaprojektowany przez firmę Oracle specjalnie dla Linuksa. Bazuje na zasadzie CoW ¹ . BtrFS wylicza sumy kontrolne dla każdego pliku, by zapewnić jego poprawność. Dodatkowo umożliwia m.in. kopiowanie przy zapisie, tworzenie migawki (możliwość powrotu do poprzedniej wersji plików), zmianę rozmiaru partycji online, kompresję w locie oraz możliwość zaadresowania 16 EiB.
Joliet	ISO9660	Wprowadzony przez Microsoft, jest rozszerzeniem systemu plików ISO 9660 i umożliwia zapis na nośnikach optycznych CD-R danych z nazwami o długości do 64 znaków — z wykorzystaniem znaków międzynarodowych, a także nazw zapisanych w standardzie UTF. Nie pozwala zapisywać plików większych niż 2 GiB.
Universal Disk Format	UDF	System plików przeznaczony do nośników optycznych BD, DVD, CD z możliwością zapisu oraz DVD-Video. Jest w pełni obsługiwany przez systemy Windows Vista, Windows 7 oraz Linux z jądrem 2.6 bez potrzeby używania dodatkowego oprogramowania.

¹ Ang. *Copy on Write* — technika kopiowania lub duplikowania danych polegająca na tym, że dane nie są fizycznie kopiowane, dopóki oryginał nie zostanie zmodyfikowany. Zwiększa to wydajność i zmniejsza zużycie zasobów poprzez ograniczenie stosowania niepotrzebnych operacji kopiowania.

System plików	Symbol	Opis
exFAT	Extended File Allocation Table (błędnie określany jako FAT64)	System plików opracowany przez firmę Microsoft specjalnie dla zewnętrznych nośników, takich jak pendrive'y i karty pamięci. Dostępny jest w systemach Windows od wersji Vista SP1. exFAT charakteryzuje się nieco większą kompatybilnością niż NTFS w przypadku urządzeń przenośnych. W porównaniu do systemu FAT32 nie ma ograniczenia wielkości pojedynczego pliku do 4 GiB (obsługuje pliki do 16 eksabajtów — 264). Oferuje wysoką wydajność przy kopiowaniu/usuwaniu plików, umożliwia tworzenie nieograniczonej liczby plików w pojedynczym katalogu.

5.5. Era systemów klient-serwer

W instytucjach i firmach coraz większą popularność zyskują systemy operacyjne działające w **strukturze klient-serwer**. Główną zasadą modelu klient-serwer jest rozproszenie elementów systemu informatycznego w obrębie różnych komputerów będących elementami sieci komputerowej. Model ten przewiduje istnienie **klientów** (stacji roboczych), czyli komputerów PC zgłaszających żądania, oraz **serwerów**, wydajnych maszyn, które obsługują zadania zgłaszane przez stacje robocze.

Serwer może pełnić funkcje: **serwera uwierzytelniającego**, **serwera plików**, **serwera wydruku**, **serwera DHCP**, **serwera DNS**, **serwera Active Directory** itd. Użytkownik zalogowany na dowolnym komputerze klienckim w sieci (w zależności od delegowanych uprawnień) może mieć dostęp do swojego konta mobilnego i korzystać z zasobów oraz mechanizmów udostępnianych przez serwer.

Istnieje grupa systemów operacyjnych, których zadaniem jest pełnienie funkcji serwerów w sieci komputerowej, np. **Novell Netware 6**, **Windows Server 2008 R2**, **2012 R2**, **2016**, **Linux z uruchomionymi aplikacjami serwerowymi**. Druga grupa to systemy działające po stronie komputerów klienckich, np. **Windows 7**, **8.1**, **10 Professional** czy **Linux**.

UWAGA

Systemy firmy Microsoft oznaczane jako Home, Basic czy Starter Edition nie są stosowane do pracy w sieci z domenami, co jednoznacznie eliminuje je z grupy systemów klienckich dla modelu Microsoft klient-serwer.