

PODZIAŁY SYSTEMÓW

Ze względu na interfejs (czyli sposób komunikowania się z użytkownikiem) systemy operacyjne dzielimy na:

- tekstowe (DOS, UNIX/Linux)
- graficzne (Windows, UNIX/Linux, MacOS)

Innym kryterium podziału systemów operacyjnych jest ilość możliwych do uruchomienia w tym samym czasie programów. Mówimy wtedy o systemach:

- jednozadaniowych – jeden proces w tym samym czasie (DOS),
- wielozadaniowych – wiele procesów w tym samym czasie (Windows, UNIX/Linux, MacOS, Novell NetWare)

Jak widać większość z dzisiejszych systemów operacyjnych jest wielozadaniowa. Cecha ta pozwala

Użytkownikowi wielozadaniowego systemu operacyjnego na np. jednoczesne przeglądanie stron internetowych, słuchanie muzyki i drukowanie dokumentów. W systemie jednozadaniowym, aby uruchomić nowy program należało zakończyć działanie poprzedniego.

Kolejnym kryterium podziału systemów operacyjnych może być jego **przeznaczenie**, czyli gdzie i do czego będzie wykorzystywany. W tym przypadku możemy wyszczególnić trzy podstawowe kategorie:

- zastosowania domowe/biurowe (Windows XP Home, Linux, MacOS)
- stacja robocza w sieci komputerowej (Windows 2000/XP Professional, UNIX/Linux)
- serwer w sieci komputerowej (Windows 2000/2003 Server, UNIX/Linux, Novell NetWare)

Z kolei pod względem architektury systemy operacyjne dzielimy na:

- monolityczne - o najprostszej strukturze i jednozadaniowe, czyli gdy system może jednocześnie wykonywać tylko jedno zadanie
- warstwowe - o hierarchicznej strukturze poleceń systemowych, system może już wykonywać w tym samym czasie kilka poleceń
- klient- serwer - o bardzo rozbudowanej strukturze, gdzie serwery pełnią nadzór nad podrzędnymi systemami zainstalowanymi w poszczególnych komputerach sieci;

Ze względu na typ jądra:

jądro monolityczne - często stosowane w systemach typu Unix. Wszystkie zadania są wykonywane przez jądro, będące jednym, dużym programem działającym w trybie jądra. Przykładami takiego jądra mogą być: Linux, OpenBSD, FreeBSD, chociaż większość posiada

umiejętność dołączania i odłączania modułów (najczęściej zawierających kod sterownika urządzenia lub obsługi potrzebnego w danej chwili systemu plików).

Zaletą tej techniki jest prostota, stabilność[potrzebne źródło], łatwość komunikacji pomiędzy różnymi członami jądra (jedna przestrzeń adresowa). Wadą jest, w późniejszym stadium rozwoju projektu, uciążliwość w rozwijaniu programu oraz w znajdowaniu błędów[potrzebne źródło].

Mikrojądro (ang. microkernel) to rodzaj jądra systemu operacyjnego, które zawiera tylko najbardziej niezbędne elementy, takie jak funkcje zarządzania wątkami, komunikacją międzyprocesową, oraz obsługą przerw i wyjątków.

Wszelkie inne zadania, takie jak np. obsługa systemów plików, sieci, sprzętu realizowane są w przestrzeni użytkownika przez osobne serwery.

Dobrymi przykładami systemów operacyjnych opartych na mikrojądrze są Amoeba, QNX, BeOS , Haiku czy Hurd, mikrojądrami są także (używane w Hurdzie) Mach i L4.

Firma Microsoft pracuje nad własnym rozwiązaniem tego typu w projekcie Singularity. Model komunikacji jądra hybrydowego z aplikacjami.

jądro hybrydowe - kompromis między architekturą jądra monolitycznego i mikrojądra. W krytycznych usługach - np. stos sieci - usługi są na stałe wkompiłowane w główny kod jądra, inne usługi pozostają oddzielone od głównego jądra i działają jako serwery (w przestrzeni jądra). Dzięki temu rozwiązaniu możliwe jest zachowanie wydajności jądra monolitycznego dla kluczowych usług. Klasyfikacja ta budzi kontrowersje niektórych programistów.

Microsoft, MacOS

POJĘCIA - SYSTEMY OPERACYJNE

API - Zbiór poleceń, za pomocą których programista wykorzystuje różne funkcje systemu operacyjnego lub uzyskuje od niego informacje. Część z tych poleceń można stworzyć samemu od podstaw (możliwość skorzystania z gotowych oszczędza jednak wiele pracy), ale istnieje także wiele poleceń API, których nie można zastąpić innymi. API na przykład pozwala bardzo szybko tworzyć graficzny interfejs użytkownika (GUI). Polecenia API są także zalecanym sposobem komunikowania się z urządzeniami, np. z drukarką. Od ang. Application Programming Interface - interfejs programowania aplikacji.

Wielozadaniowość - cecha systemu operacyjnego, która pozwala uruchomić i wykorzystywać więcej niż jedną aplikację w tym samym czasie. Można więc jednocześnie np. kopiować pliki, drukować na drukarce i dokonywać obliczeń w arkuszu kalkulacyjnym.

Wielozadaniowość z wywłaszczaniem - W trybie wielozadaniowości z wywłaszczaniem system decyduje, jak długo aplikacja korzysta z procesora i gdy ten czas upływa, przekazuje kontrolę nad nim kolejnej aplikacji (poprzednia zostaje wywłaszczona). W ten sposób zawieszony program nie może zawiesić całego systemu. Z wielozadaniowości z wywłaszczaniem korzystają systemy: OS/2, Windows 9x, Windows NT, Linux, UNIX.

Jądro systemu (kernel) - Najważniejszy komponent każdego systemu operacyjnego, wykonuje podstawowe operacje takie jak uruchamianie programów, przyznawanie zasobów

aplikacjom (odpowiada m.in. za wielozadaniowość), obsługa urządzeń oraz daty i czasu. Kernel jest ładowany do pamięci komputera jako jeden z pierwszych komponentów.

System plików - Sposób, w jaki komputer organizuje pliki i katalogi na nośniku danych. System plików określa to, jak informacje są zapisywane i odczytywane. System plików definiuje także wielkość klastrów, możliwe do użycia atrybuty plików oraz schemat poprawnych nazw plików i katalogów - ich długość i dopuszczalne znaki w nazwach. Najpopularniejsze systemy plików to FAT16 (stosowany w MS-DOS, Windows 9x i Windows NT), FAT32 (Windows 95 OSR2 i Windows 98) oraz NTFS (Windows NT). Jedynym wspólnym systemem plików wszystkich wersji Windows i MS-DOS jest FAT16. Inne popularne systemy plików to: HPFS, CDFS i VFAT.

Stronicowanie pamięci (angielskie paging),- jest to sposób zarządzania pamięcią, w którym obszar wykonywanego procesu nie musi być ciągły, dzięki czemu można unikać kosztownych reorganizacji procesów w pamięci. Poszczególne logiczne strony procesu są pomieszczone w ramkach pamięci sprzętowej, przy czym związek między stronami a ramkami jest utrzymywany za pomocą sprzętowej tablicy stron. Stronicowanie usuwa fragmentację zewnętrzną, wprowadza jednak fragmentację wewnętrzną. Stronicowanie odbywa się przy udziale sprzętu komputera.

Bitowość systemu (ilu bitowy może być system?) - Dość powszechne jest określanie systemów jako 16-bitowe (DOS, Windows), 32-bitowe (Linux, Windows NT) czy też ostatnio 64-bitowe (Windows XP). Długość rejestrów procesora, dla którego projektowany jest system, wyznacza podstawowy format wewnętrznych zmiennych, znaczników i tablic systemowych, długość ta wyznaczana jest bitach. Ze względu na szybkość działania systemu najchętniej stosowane są zmienne o długości odpowiadającej długości rejestrów procesora, co pociąga za sobą pewne konsekwencje. Na przykład, konsekwencją stosowania 16-bitowych zmiennych systemowych były ograniczenia wielkości partycji dyskowych, a w późniejszym okresie – również pojemności całych dysków. Także sposób zarządzania innymi zasobami, przede wszystkim pamięcią RAM, zależy od długości słowa procesora.

- **Wielozadaniowość** (wieloprocusowość) – możliwość wykonywania "jednocześnie" kilka procesów, otrzymywana poprzez tzw. scheduler czyli algorytm kolejujący i porządkujący procesy, które mają być wykonane. każdy proces jest wykonywany jakiś kwant czasu, a później czeka "w uśpieniu" (oczywiście z uwzględnieniem różnych priorytetów).
 - systemy wielozadaniowe :UNIX, 32-bitowe systemy z rodziny Microsoft Windows, Mac OS i jego następcą Mac OS X, AmigaOS, BeOS,
 - nie są systemami wielozadaniowymi: DOS), CP/M.
- **Wielowątkowość** możliwość wykonywania w ramach jednego procesu kilka wątków lub jednostek wykonawczych. Nowe wątki to kolejne ciągi instrukcji wykonywane oddzielnie. Wszystkie wątki tego samego procesu współdzielą kod programu i dane. W systemach nie wspierających wielowątkowości pojęcie procesu i wątku utożsamiają się.
 - systemy wielowątkowe: BeOS, Microsoft Windows 95, Windows NT, niektóre z rodziny Unix.
- **Wielobieżność** (ang. *reentrant*) to takie, w którym może pracować kilka procesów jednocześnie w trybie jądra (zapewne potrzebując jakiś funkcji systemowych).

- **Skalowalność** (ang. *scalability*) jest to cecha systemów komputerowych, polegająca na zdolności do dalszej rozbudowy, ale także miniaturyzacji systemu.

Warstwowy model oprogramowania - W modelu tym warstwę najniższą tworzą programy i procedury bezpośrednio sterujące sprzętem (ang. drivers). Korzystają z nich procedury obsługi zawarte w jądrze (Kernel), które stanowi właściwy system operacyjny.

Jądro zawiera mechanizmy systemowe, które należą do trzech grup:

- zarządzania procesami,
- zarządzania pamięcią,
- zarządzania systemem plików.

Shell – czyli powłoka realizuje (na ogół prosty) interfejs użytkownika z jądrem systemu operacyjnego, wykonuje pewne proste, wbudowane w nią polecenia wewnętrzne, jest interpretatorem specjalnego języka programowania, co umożliwia użytkownikowi programowanie w tym języku.