

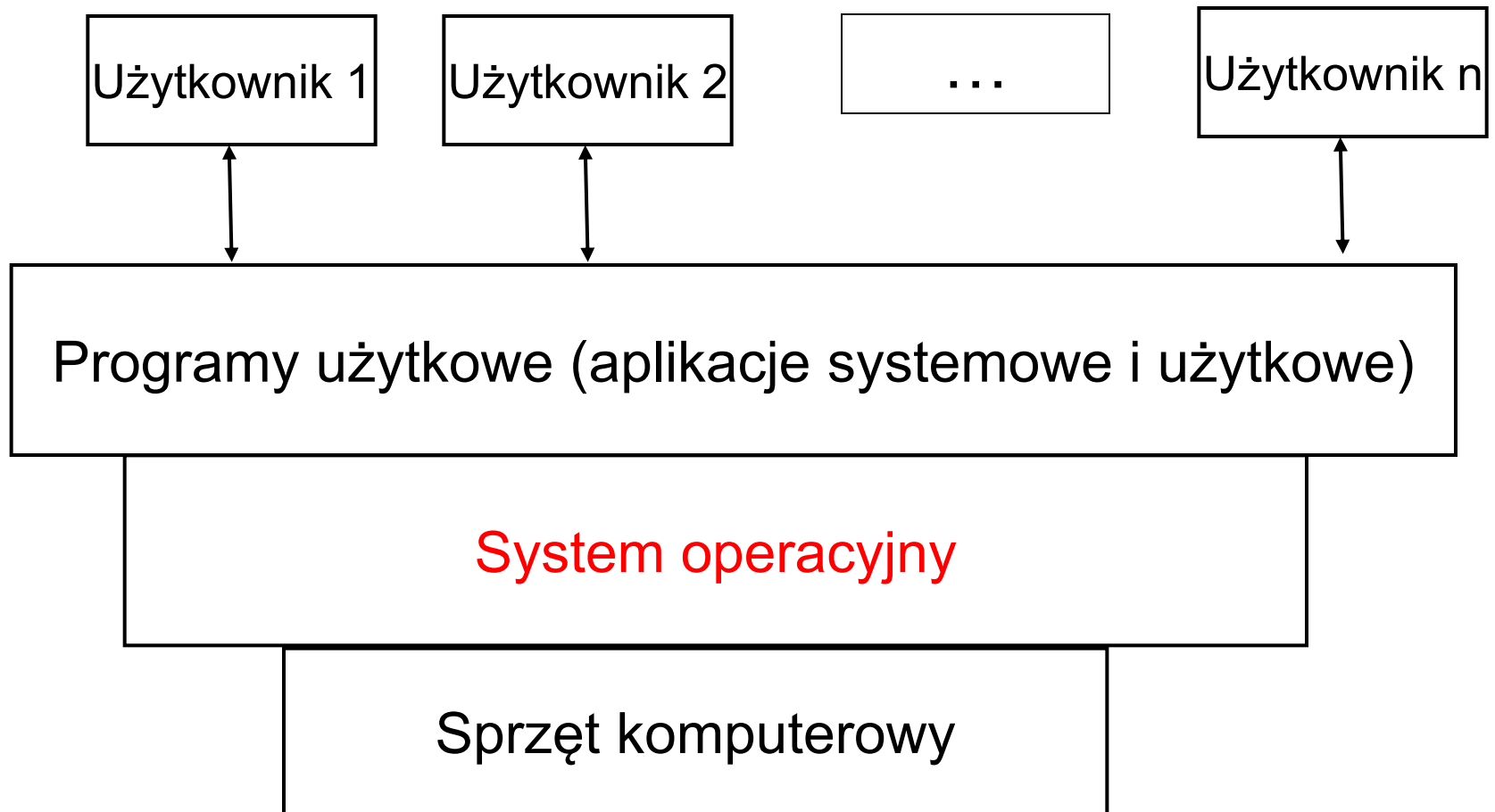
Zadania systemów operacyjnych.

System operacyjny

warstwa oprogramowania oddzielająca użytkownika od sprzętu.

- udostępnia różnorodne usługi i ma na celu ułatwienie użytkownikowi w korzystaniu z komputera.
- zwiększa wykorzystanie takich zasobów jak procesory, pamięć i urządzenia we/wy.
- tworzy środowisko dla (innych) programów.

Elementy systemu komputerowego



Zadania realizowane przez system operacyjny

Komunikacja użytkownika z komputerem
(Definicja interfejsu użytkownika).



System operacyjny dostarcza użytkownikom zbiór poleceń lub system okienkowy wraz z odpowiednim menu, który umożliwia interakcję z systemem komputerowym.

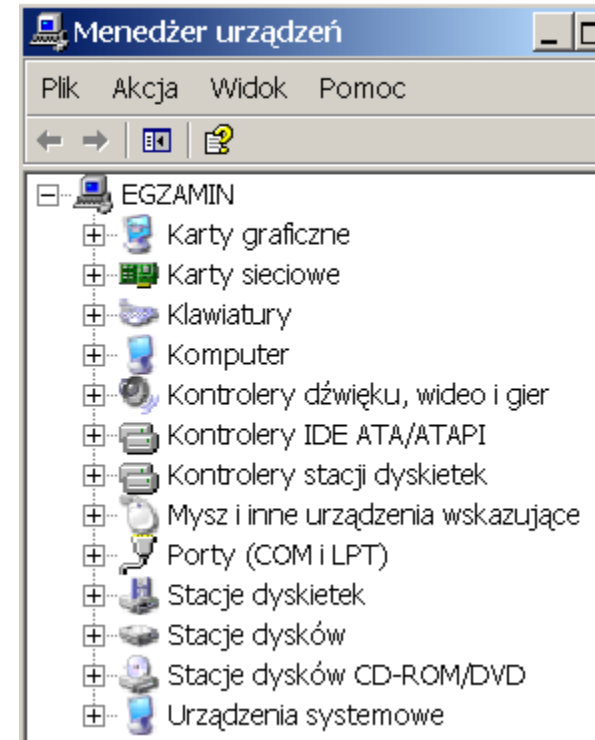
Tę rolę spełnia zewnętrzna warstwa systemu, nazywana powłoką (ang. shell), która umożliwia użytkownikowi uruchomienie aplikacji.

W środowiska graficznych do tej części systemu zalicza się dodatkowo standardowe elementy interfejsu wykorzystywane przez aplikacje np. okienka i kontrolki dialogowe, przykłady: CLI (PowerShell, Bash), GUI (Windows 11, GNOME, KDE).

Zadania realizowane przez system operacyjny

Zadania zarządzanie zasobami maszyny to:

- obsługa wewnętrznych i zewnętrznych elementów komputera takich jak: procesor, płyta główna, pamięć lub drukarka, skaner itp.
- badanie stanu urządzeń oraz poprawności komunikacji
- reakcja na błędy użytkowników (np. niedostępność zasobów, brak prawa dostępu), programistów (np. błąd dzielenie przez 0), systemu (np. błąd braku strony).
- udostępnianie urządzeń aplikacjom a tym samym użytkownikowi (np. udostępnienie drukarki w edytorze tekstu, skanera w programie graficznym, itp.),
- chwilowe i trwałe odłączanie urządzeń od komputera.



Instalowanie urządzenia polega na pobraniu sterownika danego urządzenia, przypisaniu przerwań sprzętowych (ang. IRQ), oraz kanałów do komunikacji urządzeń z pamięcią komputera (ang. DMA).

Zadania realizowane przez system operacyjny

Na czym polega instalacja urządzenia?

- **Pobranie sterownika** – oprogramowanie umożliwiające komunikację urządzenia z systemem operacyjnym.
- **Przypisanie przerw sprzętowych (IRQ)** – mechanizm obsługi sygnałów od urządzeń.
- **Przydzielenie kanałów DMA (Direct Memory Access)** – bezpośrednia komunikacja urządzenia z pamięcią RAM bez obciążania CPU.

Nowoczesne mechanizmy instalacji

- **Plug and Play (PnP)** Automatyczne wykrywanie i konfiguracja urządzeń w systemie Windows i Linux.
- **Automatyczne aktualizacje sterowników** Wbudowane w Windows Update, narzędzia producentów (np. NVIDIA GeForce Experience).
- **udev (Linux)** Dynamiczne zarządzanie urządzeniami w przestrzeni użytkownika, automatyczne tworzenie plików urządzeń.

Zadania realizowane przez system operacyjny

Komunikacja z innymi maszynami

- Moduły systemu operacyjnego obsługują sieci komputerowe, umożliwiając dostęp do LAN, WAN i Internetu.
- Klasyczne funkcje: konfiguracja IP, obsługa protokołów (TCP/IP, DHCP, DNS).
- Nowoczesne funkcje:**
 - Obsługa chmury (Azure, AWS, Google Cloud) – integracja z usługami wirtualnymi, backup, synchronizacja.
 - Konteneryzacja (Docker, Kubernetes) – izolacja aplikacji, skalowalność, automatyzacja wdrożeń.

Zadania realizowane przez system operacyjny

Zadania realizowane przez system operacyjny

- Zarządzanie sprzętem (CPU, RAM, urządzenia I/O).
- Zarządzanie plikami i systemem plików.
- Zarządzanie procesami i pamięcią.
- Nowoczesne zadania:
 - Integracja z chmurą.
 - Obsługa wirtualizacji i konteneryzacji.
 - Automatyzacja (PowerShell, Ansible).
 - Bezpieczeństwo (SELinux, AppArmor, Windows Defender).

Zadania realizowane przez system operacyjny

Zadania w zakresie programów użytkowych (aplikacje systemowe i użytkowe)

- zarządzanie plikami - system operacyjny organizuje i ułatwia dostęp do informacji (zwłaszcza gromadzonej w pamięci masowej) np. w postaci hierarchicznego systemu plików,
- uruchamianie aplikacji - system operacyjny dostarcza struktur danych do organizacji wykonywania programu oraz zachowywania i odtwarzania stanu przetwarzania (procesy i przełączanie kontekstu). System operacyjny udostępnia też programistom mechanizmy komunikacji pomiędzy procesami (kolejki komunikatów, strumienie, pamięć współdzielona) oraz mechanizmy synchronizacji procesów (np. semafony, monitory).

Funkcje systemu operacyjnego

- planowanie oraz przydział czasu procesora poszczególnym zadaniom (procesom) - jest to zadanie jądra,
- kontrola i przydział pamięci operacyjnej dla zadań, które uruchamia,
- synchronizacja zadań i komunikacja pomiędzy zadaniami,
- obsługa sprzętu oraz zapewnienie równoległe wykonywanym zadaniom dostępu do sprzętu bez interferencji (czyli wolne od procesów wykorzystujących ten sam zasób w tym samym czasie),
- dostarczenie środowiska dla programów użytku codziennego, np. do pracy,

Funkcje systemu operacyjnego

- dzielenie się zasobami wśród wielu użytkowników,
- sterowanie urządzeniami wejście i wyjścia, np. klawiaturą, myszą, skanerem,
- wykorzystanie sprzętu komputerowego,
- dostarczenie wygodnego interfejsu dla użytkownika (wiele systemów operacyjnych posiada środowiska graficzne ułatwiające komunikację maszyny z użytkownikiem),
- ustalanie połączeń sieciowych, zarządzanie plikami (może, ale nie musi się zajmować).

Oprogramowanie systemowe zadania

Oprogramowanie systemowe (System software) – kontroluje i koordynuje działanie zasobów sprzętowych przez różne programy użytkowe dla różnych użytkowników.

Klasyczne funkcje oprogramowania systemowego

1. Zarządzanie sprzętem. Obsługa procesora, pamięci, urządzeń wejścia/wyjścia.

Przykład:

Windows 11 / Windows Server 2022:

- Obsługa wielordzeniowych procesorów i technologii Hyper-Threading.
- Zarządzanie energią (ACPI), tryby oszczędzania energii.
- Dynamiczne zarządzanie urządzeniami przez Plug and Play.

Linux (Ubuntu, RHEL):

- udev – dynamiczne wykrywanie i konfiguracja urządzeń.
- Obsługa nowoczesnych magistrali (PCIe, NVMe).
- Zarządzanie CPU i I/O schedulerami (np. cfq, deadline).

Hypervisory (VMware, Hyper-V):

- Wirtualizacja sprzętu (CPU, pamięć, urządzenia).
- Dynamiczne przydzielanie zasobów dla maszyn wirtualnych.

Sterowniki i firmware:

- Automatyczne aktualizacje przez Windows Update lub Linux kernel modules.

Oprogramowanie systemowe zadania

Oprogramowanie systemowe (System software) – kontroluje i koordynuje działanie zasobów sprzętowych przez różne programy użytkowe dla różnych użytkowników.

Klasyczne funkcje oprogramowania systemowego

2. Zarządzanie plikami. Tworzenie, odczyt, zapis, usuwanie plików. Przykład:

Windows:

- NTFS – obsługa uprawnień, szyfrowania (EFS), kompresji.
- ReFS – odporność na uszkodzenia, integracja z serwerami.

Linux:

- ext4 – journaling, obsługa dużych wolumenów.
- XFS, Btrfs – migawki, sumy kontrolne, dynamiczne zarządzanie.

macOS:

- APFS – szyfrowanie, klonowanie plików, snapshoty.

Dodatkowe funkcje:

- Obsługa uprawnień ACL.
- Mechanizmy szyfrowania (BitLocker, LUKS).
- Integracja z chmurą (OneDrive, Google Drive, Dropbox).

Oprogramowanie systemowe zadania

Oprogramowanie systemowe (System software) – kontroluje i koordynuje działanie zasobów sprzętowych przez różne programy użytkowe dla różnych użytkowników.

Klasyczne funkcje oprogramowania systemowego

3. Zarządzanie procesami

Windows 11 / Windows Server 2022:

- Task Scheduler – planowanie zadań.
- Task Manager – monitorowanie i kontrola procesów.
- Obsługa wielowątkowości i priorytetów procesów.

Linux (Ubuntu, RHEL):

- systemd – zarządzanie usługami i procesami.
- Polecenia: ps, top, kill, nice, renice.
- Obsługa kontenerów (Docker, Podman) – izolacja procesów.

Nowoczesne mechanizmy:

- cgroups – ograniczanie zasobów dla procesów.
- Namespaces – izolacja środowisk.
- Hypervisory – zarządzanie procesami w maszynach wirtualnych.

Oprogramowanie systemowe zadania

Oprogramowanie systemowe (System software) – kontroluje i koordynuje działanie zasobów sprzętowych przez różne programy użytkowe dla różnych użytkowników.

Klasyczne funkcje oprogramowania systemowego

1. Zarządzanie pamięcią

Przydzielanie pamięci programom, obsługa pamięci wirtualnej.

Przykład: Windows 11 / Windows Server 2022: Dynamiczne przydzielanie pamięci, Linux (Ubuntu, RHEL): swap + zram – kompresja pamięci wirtualnej, HugePages – optymalizacja dla dużych baz danych.

2. Obsługa urządzeń i sterowników

Komunikacja z drukarkami, kartami sieciowymi, dyskami.

Przykład: Windows Plug and Play (PnP) – automatyczne wykrywanie i instalacja sterowników. Driver Booster, Windows Update – aktualizacja sterowników.

3. Interfejs użytkownika

Tekstowy (CLI) lub graficzny (GUI).

Przykład: PowerShell (Windows), Windows 11 (GUI).

Zadania warstw systemu komputerowego

Oprogramowanie systemowe (System software) – kontroluje i koordynuje działanie zasobów sprzętowych przez różne programy użytkowe dla różnych użytkowników.

Nowoczesne funkcje:

- Obsługa wirtualizacji (Hyper-V, VMware) – umożliwia uruchamianie wielu systemów na jednym sprzęcie.
- Integracja z chmurą (Azure, AWS) – pozwala na pracę w środowiskach hybrydowych.
- Mechanizmy bezpieczeństwa (SELinux, AppArmor, szyfrowanie) – ochrona danych i kontrola dostępu.
- Automatyzacja i zarządzanie konfiguracją (PowerShell, Ansible) – ułatwia administrację i wdrożenia.



Zadania warstw systemu komputerowego

Oprogramowanie narzędziowe (Software tools) – udostępnia dogodne interfejsy użytkowe wspomagające zarządzanie zasobami sprzętowymi oraz usprawniające działanie systemu.

Klasyczne przykłady:

Edytory tekstu – np. Notepad, Vim.

Kompilatory – GCC, Turbo Pascal.

Debuggery – GDB.

Systemy kontroli wersji – CVS, Subversion.

Narzędzia do archiwizacji – WinRAR, tar.

Nowoczesne przykłady:

Wirtualizacja – VMware, Hyper-V.

Konteneryzacja – Docker, Kubernetes.

Narzędzia DevOps – Jenkins, Ansible, Terraform.

Systemy CI/CD – GitLab CI, GitHub Actions.

Monitorowanie i zarządzanie – Prometheus, Grafana.



Zadania warstw systemu komputerowego

Oprogramowanie użytkowe (Utility software) – określa sposoby użycia zasobów systemowych do rozwiązywania problemów obliczeniowych zadanych przez użytkownika.

Klasyczne przykłady:

- Edytory tekstu – WordPerfect, Microsoft Word (starsze wersje).
- Arkusze kalkulacyjne – Lotus 1-2-3, Excel.
- Programy graficzne – CorelDRAW, Paint.
- Systemy baz danych – dBase, MS Access.
- Programy do tworzenia prezentacji – Harvard Graphics.

Nowoczesne przykłady:

- Chmura obliczeniowa – Microsoft Azure, AWS, Google Cloud.
- Systemy baz danych – MySQL, PostgreSQL, MongoDB.
- Aplikacje biurowe online – Microsoft 365, Google Workspace.
- Narzędzia do analizy danych – Power BI, Tableau.
- Aplikacje komunikacyjne – Teams, Slack.



Zadania warstw systemu komputerowego

Kategorie użytkowników systemów informatycznych

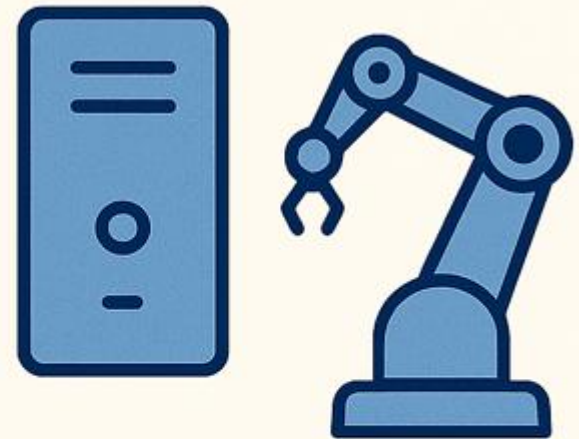
1. Użytkownicy końcowi – osoby korzystające z aplikacji (np. pakiet biurowy, CRM).
2. Administratorzy – zarządzają konfiguracją i bezpieczeństwem.
3. Maszyny i urządzenia – np. sterowniki PLC, IoT.
4. Systemy i komputery – komunikacja w sieciach, integracja usług.



Ludzie



Administratorzy



Maszyny
i komputery

Pytania

1. Jaką rolę pełni oprogramowanie systemowe?
2. Jakie są klasyczne funkcje oprogramowania systemowego?
3. Podaj współczesne odpowiedniki dla: zarządzania sprzętem, plikami, procesami, pamięcią, sterownikami, interfejsem użytkownika.
4. Jakie mechanizmy bezpieczeństwa stosują systemy operacyjne?
5. Czym różni się oprogramowanie narzędziowe od użytkowego?
6. Podaj przykłady oprogramowania użytkowego.
7. Jakie są przykłady wirtualizacji i konteneryzacji?
8. Podaj przykłady platform chmurowych.
9. Jakie narzędzia wspierają DevOps?
10. Kim są użytkownicy systemu komputerowego?
11. Dlaczego interfejs urządzeń wejścia-wyjścia jest istotny?
12. Jakie są przykłady komunikacji przewodowej i bezprzewodowej w systemie komputerowym?