Systemy operacyjne

Wprowadzenie

2019/2020

Jarosław Koźlak

Prowadzący wykład

- **Prowadzący:**
  - dr hab. inż. Jarosław Koźlak

- **Kontakt:**
  - e-mail: kozlak@agh.edu.pl
  - www: http://home.agh.edu.pl/~kozlak
  - telefon: … 12 3283308
  - pokój: : D17, 1 piętro, pokój 2.24

Organizacja wykładu i egzaminu

- **Lab**
  - dopuszczalna jedna nieobecność nieusprawiedliwiona

- **Egzamin:**
  - ocena 5.0 z laboratorium dopuszcza do zerowego terminu egzaminu

Tematyka wykładów

1. Wprowadzenie. Historia systemów operacyjnych. Struktura systemów komputerowych i systemów operacyjnych.
2. Zarządzanie procesami: Procesy, planowanie przydziału procesora
3. Zarządzanie procesami: Synchronizacja.
4. Zarządzanie procesami: Zakleszczenia
5. Komunikacja między procesami
6. Zarządzanie pamięcią: informacje wstępne, wymiana
7. Zarządzanie pamięcią: pamięć wirtualna, algorytmy wymiany stron, segmentacja.
8. Biblioteki statyczne i dynamiczne
9. Realizacja systemu plików
10. Urządzenia i systemy wejścia-wyjścia
11. Przykładu systemów: Unix i Linux
12. Przykłady systemów: Windows NT

Literatura. Teoria

1. **Abraham Silberschatz, Peter B. Galvin, "Podstawy systemów operacyjnych", WNT, 2005 (wyd. 6 zmienione)**
   - **A. Silberschatz, P.B. Galvin, G. Gagne. „Operating System Concepts. Eighth Edition”, John Wiley & Sons Inc, 2009**

2. **William Stallings, „Systemy operacyjne”, 2018**
   - **William Stallings, „Operating Systems. Internals and Design Principles”. Ninth Edition, Pearson Prentice Hall, 2017**
3. Andrew S. Tanenbaum, "Modern Operating Systems", Third Edition, Prentice Hall International, 2008
4. Andrew S. Tanenbaum, Albert S. Woodhull, "Operating Systems. Design and Implementation" Prentice-Hall International, Inc. , 3rd ed, 2006
5. Thomas W. Doeppner, „Operating Systems in Depth”, Wiley, 2010
6. Thomas Anderson, Michael Dahlin, „Operating Systems. Principles & Practice”, Second Edition, 2011-2014

Literatura. Programowanie współbieżne

1. Ben-Ari “Podstawy programowania współbieżnego i rozproszonego”, Wydawnictwo Naukowo-Techniczne, Warszawa 1996.
2. Iszkowski, M. Maniecki “Programowanie współbieżne`”, Wydawnictwo Naukowo-Techniczne, Warszawa 1982.
3. Z. Weiss, T. Gruźlewski “Programowanie współbieżne i rozproszone w przykładach i zadaniach, Wydawnictwo Naukowo-Techniczne, Warszawa 1994.

Architektura systemu Unix/Linux

1. **Berny Goodheart, James Cox, "Unix System V Wersja 4 od środka", Wydawnictwo Naukowo-Techniczne, 2001 ("The magic garden explained. The internals of Unix System V Release 4. An open systems design", 1994)**

2. **Uresh Vahalia, „Jądro systemu Unix...”, WNT 2001**

3. **Daniel P. Bovet, Marco Cesati, „Understanding the Linux Kernel”, Third edition, O’Reilly, 2006**
   - Daniel P. Bovet, Marco Cesati, "Linux Kernel", Wydawnictwo RM 2001

4. **Robert Love, „Linux Kernel Development”, Novel Press, Third edition, 2010**

Programowanie w systemie Unix

1. **W. Richard Stevens, „Programowanie w środowisku systemu Unix”, WNT, 2002.**
   W. Richard Stevens, Stephen A. Rago, "Advanced programming in the Unix Environment", Third Edition, Addison Wesley Publishing Company, 2013
2. **Mark Mitchell, Jeffrey Oldham, Alex Samuel, "Linux. Programowanie dla zaawansowanych", Wydawnictwo ReadMe, 2002 (orig: "Advanced Linux Programming", New Riders Publishing, 2001)**
3. W. Richard Stevens, “Unix: Programowanie sieciowe”, t1-t2 WNT, 2000,2001
4. Kay A. Robbins, Steven Robbins, „Unix System Programming. Communication, concurrency and threads”, Prentice Hall, 2003
5. Marc J. Rochkind, „Advanced UNIX Programming”, Addison-Wesley, Second edition, 2004
6. **Michael Kerrisk, A Linux and Unix System Programming Handbook, no starch press. 2010.**

Użytkowanie systemu Unix/Linux.

1. Matt Welsh, Matthias Kalle Dalheimer, Lar Kaufman, "Linux", Wydawnictwo RM, 2002 (orig: "Running Linux", third edition, 1999)
2. M. Tim Jones, GNU/Linux Application Programming, Charles River Media, INC, 2005

Windows

1. William R. Stanek, „Windows Server 2008. Inside Out,” Microsoft Press, 2008

2. Charles Petzold, „Programowanie Windows”, RM (orig: Microsoft Press),1999
3. Victor Toth , "Programowanie Windows 98/NT", Helion 1999
4. Jeffrey Richter, Programowanie aplikacji dla Microsoft Windows, Wydawnictwo RM, 2002

5. Mark E. Russinovich, David A. Solomon, Alex Ionescu, „Windows Internals” Part 1-2, Microsoft Press, 2012

Wybrane strony WWW

- Levenez E.: *UNIX History,* http://www.levenez.com/unix/
- Levenez E.: *Windows History,* http://www.levenez.com/windows/
- The Open Group: *UNIX History and Timeline*, http://www.unix.org/what_is_unix/history_timeline.html

- theForger's Win32 API Tutorial, http://www.winprog.org/tutorial/

- POSIX Threads Tutorial, http://math.arizona.edu/~swig/documentation/pthreads/
- POSIX Threads Programming, https://computing.llnl.gov/tutorials/pthreads/

- The Linux Kernel Archives, http://kernel.org/
- Linux Documentation Project, http://mirrors.kernel.org/LDP/

Regulamin laboratorium

Zaliczenie laboratorium uzyskuje się po zrealizowaniu następujących warunków:

- uczęszczanie na laboratoria i realizacja zadanych na nich zadań, bierze się pod uwagę:
  - obecność (dopuszczalna jedna nieobecność nieusprawiedliwiona),
  - przygotowanie do zajęć,
  - realizacja zadań (dopuszczalny 1 tydzień spóźnienia)

- zaliczenie sprawdzianów (tematyka ostatnich ćwiczeń) i 2 x kolokwium (tematyka ostatnich kilku ćwiczeń)

Inne uwagi:

- podział na grupy, grupy zrównoważone

Plan laboratorium

1. g++/gcc. Make i gdb. Zarządzanie pamięcią. Biblioteki statyczne i dynamiczne. Przydatne funkcje. Pomiar czasu
2. Operacje na plikach
3. Tworzenie procesów. Środowisko procesu. Sterowanie procesami.
4. Sygnały
5. Potoki nazwane i nienazwane
6. Kolejki komunikatów (IPC & Posix)
7. Semafory i pamięć wspólna (IPC & Posix)
8. Wątki (podstawy)
9. Wątki (synchronizacja)
10. Sokety

Wprowadzenie

Struktura systemu komputerowego

System komputerowy złożony ze sprzętu, programów systemowych i aplikacji.

Struktura systemu komputerowego 2

- **Urządzenia fizyczne :** zintegrowane układy elektroniczne, kable danych, kable zasilające itd.
- **Mikroprogram:** prosty software, który bezpośrednio kontroluje urządzenia fizyczne, zwykle jest ulokowany w pamięci ROM
- **Język maszynowy:** obejmuje zbiór instrukcji, które mikroprogram potrafi wykonać, typowo 50-300 instrukcji dotyczących przesyłania danych, operacji arytmetycznych, porównywania wartości. Urządzenia WE/WY są obsługiwane przez ładowanie wartości do ***rejestrów urządzeń***
- **System operacyjny:** ma ukrywać złożoność operacji na sprzęcie i dostarczać wygodniejszy zbiór instrukcji. SO jest częścią oprogramowania, które jest uruchamiane w trybie jądra/trybie użytkownika
- **Powyżej systemu operacyjnego:** reszta oprogramowania i aplikacje, uruchamiane w trybie użytkownika.

Co to jest system operacyjny **?**

- **Popularne określenie zakresu systemu operacyjnego:**
  - To, co dostawca przysyła w odpowiedzi na nasze zamówienie „systemu operacyjnego”

- System operacyjny – przypomina rząd: Nie wykonuje sam żadnej użytecznej funkcji, ale ma za zadanie przygotować środowisko, w którym inne programy mogą wykonywać pożyteczne funkcje.

Różne spojrzenia na system operacyjny

- **System operacyjny jako Rozszerzona Maszyna**
  - architektura komputerów na poziomie języka maszynowego jest trudna do użycia w programach (szczególnie: operacje wejścia/wyjścia)
  - zadaniem systemu operacyjnego jest ukrycie tej złożoności i dostarczenie programiście bardziej przyjaznego interfejsu
  - system operacyjny udostępnia maszynę rozszerzoną /maszynę wirtualną, łatwiejszą do programowania

- **System Operacyjny jako Zarządca Zasobów**
  - różne rodzaje zasobów w systemie: procesory, pamięć, zegary, dyski, terminale, napędy taśmy magnetyczne, drukarki itd.
  - komputer ma za zadanie udostępniać zasoby użytkownikowi, kontrolować dostęp do nich i zapobiegać chaosowi i konfliktom między programami (i użytkownikami)

- **System operacyjny jako program sterujący:**
  - Nadzoruje działanie programów użytkownika, przeciwdziała błędom i zapobiega niewłaściwemu użyciu komputera.

Cele systemu operacyjnego

- wygoda użytkownika (!)
- efektywne działanie systemu komputerowego (szczególnie istotne w rozbudowanych, wielodostępnych systemach z podziałem czasu)

- możliwe sprzeczności między powyższymi celami

- początkowo: przedkładano wydajność nad wygodę
- obecnie: najczęściej przedkłada się wygodę nad wydajność

Historia systemów operacyjnych: Proste systemy wsadowe

Cecha charakterystyczna: brak bezpośredniego nadzoru ze strony użytkownika podczas wykonywania jego zadania

Struktura systemu komputerowego:

- wielkie fizycznie maszyny obsługiwane przez konsolę
- urządzenia wejściowe: czytniki kart i przewijaki taśm
- urządzenia wyjściowe: drukarki wierszowe, przewijaki taśm, perforatory kart
- użytkownik przygotowywał zadanie (program, dane i informacje sterujące zapisane na kartach) i przekazywał je operatorowi
- wyniki były uzyskiwane po pewnym czasie (min., godz., dni)

Historia systemów operacyjnych: Proste systemy wsadowe 2

Struktura systemu operacyjnego:

- SO rezyduje na stałe w pamięci operacyjnej
- obowiązek SO : automatyczne przekazywanie sterowania od jednego zadania do następnego
- w celu przyspieszenia przetwarzania, zadania o podobnych wymaganiach grupowane razem i wykonywane w formie tzw. wsadu (batch), sortowanie dokonywane przez operatora
- jednostka centralna często pozostaje bezczynna w wyniku wolnej pracy mechanicznych urządzeń WE/WY

Historia systemów operacyjnych: Proste systemy wsadowe

Modyfikacje:

- wprowadzono dyski , co pozwoliło na zastosowanie spooling'u (Simultaneous Peripheral Operation On-Line – jednoczesna, bezpośrednia praca urządzeń)
- spooling pozwala na jednoczesne wykonywanie obliczeń jednego zadania i operacji WE/WY innego
- dysk – bufor, pozwala na czytanie z maksymalnym wyprzedzeniem z urządzeń WE i przechowywanie plików wyjściowych, aż urządzenia WY będą je mogły przyjąć
- zawartość karty nie była po odczytaniu z czytnika ładowana bezpośrednio do pamięci, ale przechowywana na dysku, a system operacyjny przechował tablicę opisującą rozmieszczenie obrazów kart na dysku
- podobnie – komunikaty wyjściowe były zapisywane do bufora systemowego i na dysku, a drukowane dopiero po zakończeniu zadania

Historia systemów operacyjnych: Wieloprogramowane systemy wsadowe

- Ze spoolingiem jest związana pula zadań (**job pool**): zadania, które zostały odczytane na dysk, gdzie czekają na wykonanie
- Istnieje możliwość **wyboru** przez system zadania do wykonania, spośród zadań przechowywanych na dysku.
- Wybór zadania jest związany z użyciem planowania zadań (szeregowania zadań, ang. job scheduling)
- Najważniejszy aspekt planowania zadań: **wieloprogramowanie**
- **Wieloprogramowanie:** w tym samym czasie system operacyjny przechowuje w pamięci kilka zadań – część z zadań zgromadzonych w **puli zadań**
- Przesłanki stosowania wieloprogramowania: jedno zadanie jednego użytkownika na ogół nie jest w stanie utrzymać cały czas w aktywności procesora lub urządzeń WE/WY

Historia systemów operacyjnych: Systemy z podziałem czasu

Wady systemów wsadowych:

- użytkownik nie może ingerować w zadanie podczas jego wykonywania się, musi zatem przygotować karty sterujące dla wszystkich możliwych zdarzeń.
- następne kroki wykonania zadania mogą zależeć od wcześniejszych wyników (np. kompilacja, uruchamianie...)
- trudności w testowaniu, programista nie może na bieżąco zmieniać programu w celu obserwacji jego zachowań

Wielozadaniowość i systemy z podziałem czasu

- **Wielozadaniowość (ang. multitasking):** procesor wykonuje na przemian wiele zadań, przełączenia między nimi występują tak często, że użytkownicy mogą mogą współdziałać ze swymi programami podczas ich wykonania.
- **System z podziałem czasu (ang. time-sharing systems)** – wielu użytkowników, każdy uzyskuje dostęp do procesora przez pewną małą porcję czasu; każdy użytkownik ma przynajmniej jeden proces w pamięci
- **Bezpośredni dostęp do systemu plików** (ang. on-line file system) – umożliwia wygodne korzystanie z danych i oprogramowania.
- **Plik (ang. file)** – zestaw powiązanych informacji, zdef. prze twórcę
- **Interakcyjny/bezpośredni (ang. hands on)** system komputerowy umożliwia bezpośredni dialog użytkownika z systemem
- Wejście (zazwyczaj): klawiatura,
- Wyjście (zazwyczaj): ekran (np. monitora)
- „instrukcje sterujące” przekazywane ze pośrednictwem klawiatury
- **Czas odpowiedzi** (ang. response time) powinien być krótki – max. **rzędu sekund**

Systemy wsadowe, a systemy interakcyjne

- systemy wsadowe są odpowiednie dla wielkich zadań, których wykonanie nie wymaga bezpośredniego dozoru
- zadanie interakcyjne składa się z wielu krótkich działań, a rezultaty poszczególnych poleceń mogą być nieprzewidywalne
- Rozbudowane mechanizmy: kolejkowanie, pamięć wirtualna itd.

Systemy operacyjne dla komputerów osobistych

- spadek cen sprzętu komputerowego -> stworzenie (znów) systemów komputerowych dla indywidualnych użytkowników (lata 70-te)
- Zmiany urządzeń WE/WY:
  - pulpity przełączników, czytniki kart -> klawiatury (wzór. na maszynach do pisania) i myszki
  - drukarki wierszowe i perforatory kart -> monitory ekranowe i małe, szybkie drukarki
- początkowo procesory są pozbawione cech potrzebnych do ochrony systemu operacyjnego przed programami użytkowymi, systemy operacyjne nie są wielostanowiskowe ani wielozadaniowe
- w rozwoju położono nacisk na maksimum wygody użytkowania i szybkość kontaktu z użytkownikiem
- z czasem –ze wzrostem możliwości sprzętu i podłączeniem mikrokomputerów do sieci- początkowo cechy dużych maszynach, są adoptowane do mikrokomputerów (ochrona plików, wielozadaniowość)

Systemy równoległe

- systemy wieloprocesorowe (systemy ściśle powiązane)
- **systemy wieloprocesorowe (ang. multiprocessor systems) –** pewna liczba procesorów współpracuje ze sobą dzieląc szynę, zegar i (czasami) pamięć i urządzenia zewnętrzne
- **takie systemy: systemy ściśle powiązane (tightly coupled)**
- **Zalety systemów wieloprocesorowych:**
  - **przyspieszenie pracy** (ale przy n procesorach < n)
  - **zwiększenie niezawodności**
  - **fault tolerant systems – systemy tolerujące awarie**
  - **właściwość łagodnej degradacji (ang. graceful degradation) --** system jest w stanie kontynuować pracę po awarii części sprzętu

Modele działania systemów wieloprocesorowych:

sprzętowe i programowe podwojenie funkcji

- – 2 identyczne procesory z lokalną pamięcią: procesor podstawowy i zapasowy, każdy proces ma 2 kopie na maszynie podstawowej i zapasowej, w ustalonych punktach kontrolnych stan informacji o każdym zadaniu jest kopiowany z maszyny podst. do zapasowej; kosztowne rozwiązanie

wieloprzetwarzanie symetryczne (ang. symmetric multiprocessing) –

- na każdym procesorze działa identyczna kopia systemu operacyjnego, które komunikują się w zależności od potrzeb:
- zaleta: równocześnie może pracować wiele procesów, bez pogarszania działania całego systemu
- wada: może dojść do sytuacji, że jedne procesory będą przeciążone, a inne – słabo obciążone, dlatego procesory mogą korzystać z pewnych wspólnych struktur danych; np. Solaris

wieloprzetwarzanie asymetryczne (ang. asymmetric multiprocessing) –

- każdy procesor ma przypisane inne zadanie, istnieje wyróżniony procesor główny, który zarządca systemem – najczęściej występuje w bardzo wielkich systemach, w których najwięcej czasu zajmują oper WE/WY -> zastosowanie procesora czołowego (front-end processor), który działa jak bufor między końcówką konwersacyjną i procesorem głównym;

Systemy rozproszone (ang. distributed systems)

inna nazwa: luźno połączone (ang. loosely coupled)

- wiele procesorów, procesory nie dzielą pamięci ani zegara
- każdy procesor ma własną pamięć lokalną
- komunikacja między procesorami przy użyciu linii komunikacyjnych
- procesory w takim systemie mogą być nazywane: stanowiska (sites), węzły (nodes),

Przyczyny tworzenia systemów rozproszonych:

- **podział zasobów –** użytkownik jednego stanowiska może korzystać z zasobów (np. plików, drukarek) dostępnych na innych
- **przyspieszanie obliczeń**
  - jeżeli obliczenie można rozłożyć na zbiór obliczeń cząstkowych, które mogą być wykonywane współbieżnie, to można je przydzielić do poszczególnych stanowisk
  - jeżeli stanowisko jest przeciążone zadaniami, to część z nich można przenieść do innego, mniej obciążonego – dzielenie obciążeń (ang. load sharing)
- **niezawodność --** w przypadku awarii jednego stanowiska, pozostałe mogą kontynuować pracę
- **komunikacja –** możliwość wymiany informacji, okna (X), poczta elektroniczna (ang. electronic mail),

Systemy czasu rzeczywistego (ang. real time systems)

- stosowane, gdzie istnieją surowe wymagania na czas wykonania operacji lub przepływu danych

- Przykład: sterownik w urządzeniu np. przy nadzorowaniu eksperymentów naukowych, obrazowaniu badań medycznych, sterowaniu procesami przemysłowymi , systemach wizualizacji
- komputer pozyskuje dane z czujników o musi je analizować i regulować działanie kontrolowanego obiektu, w zależności od jego stanu

- **Wymaganie systemu czasu rzeczywistego:** przetwarzanie danych **musi** się zakończyć przed upływem określonego czasu

- **W systemie z podziałem czasu:** szybkie uzyskanie odpowiedzi jest pożądane (ale nie jest konieczne)
- **W systemie wsadowym:** nie ma ograniczeń czasowych

Rodzaje systemów czasu rzeczywistego

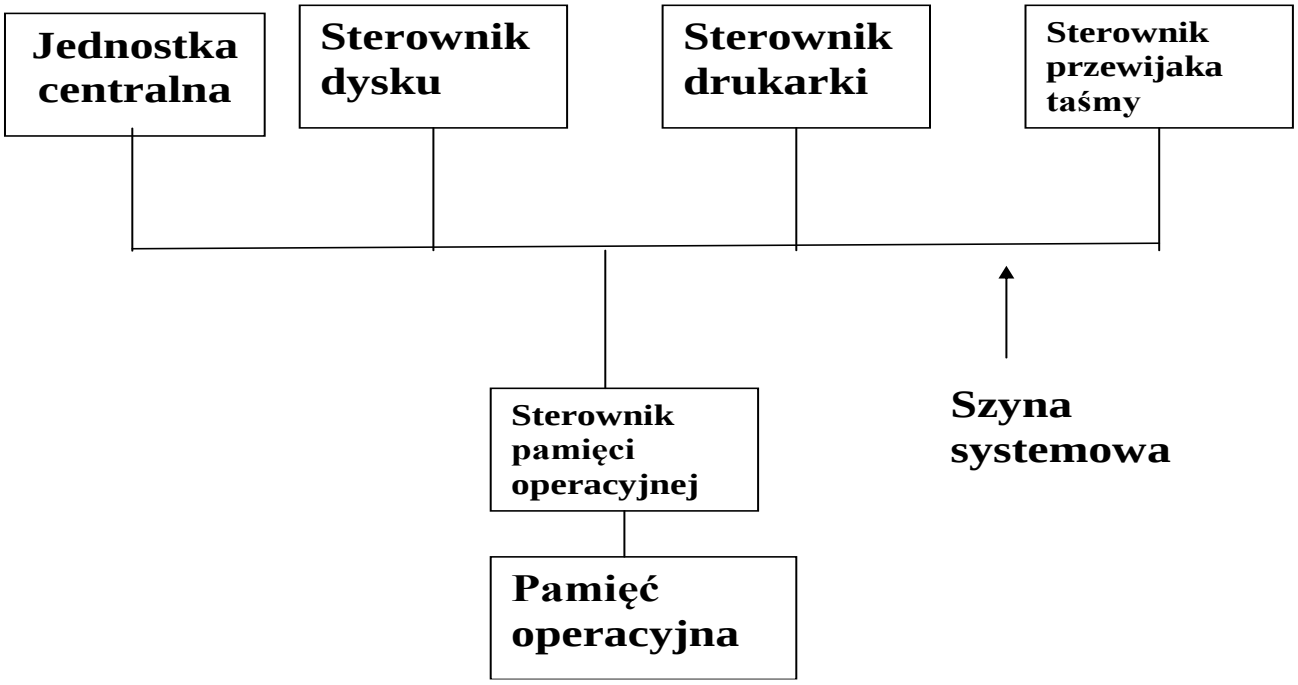
- **rygorystyczny system czasu rzeczywistego (ang. hard real-time system) –** gwarantuje terminowe wypełnianie krytycznych zadań, systemy o specjalnej konstrukcji np. dane w pamięci o krótkim czasie dostępu lub w pamięci ROM, z reguły brak pamięci wirtualnej.
- Żaden z istniejących, uniwersalnych systemów operacyjnych nie umożliwia działania w czasie rzeczywistym.

- **łagodny system czasu rzeczywistego (ang. soft real time system) –** krytyczne zadanie do obsługi w czasie rzeczywistym otrzymuje pierwszeństwo przed innymi zadaniami i zachowuje je aż do swojego zakończenia, opóźnienia muszą być ograniczone --- zadanie nie może czekać w nieskończoność na usługi jądra
- zastosowanie w przemyśle i robotyce jest ryzykowne, przydatne w technikach multimedialnych, kreowaniu sztucznej rzeczywistości, zaawansowanych projektach badawczych (wyprawy planetarne, badania podmorskie)
- Większość współczesnych systemów operacyjnych (w tym Unix) może spełniać te wymagania

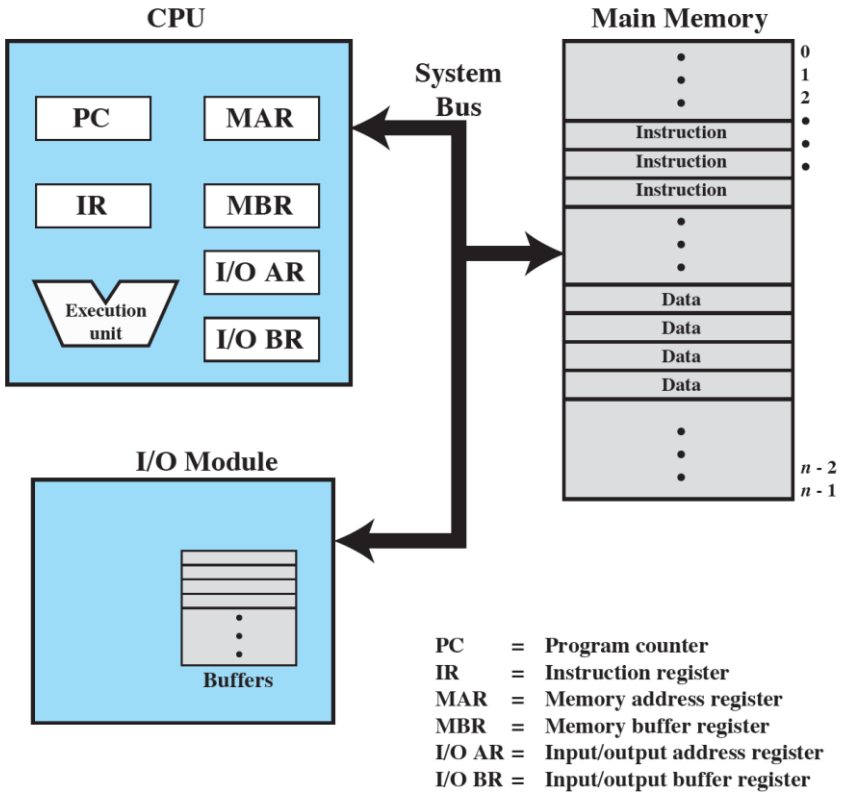
Struktura systemów komputerowych

Elementy składowe systemu komputerowego:

- jednostka centralna (ang. central processor unit - CPU)
- pewna liczba sterowników urządzeń (ang. device controllers)
- pamięć operacyjna
- szyna systemowej łączącej poszczególne elementy

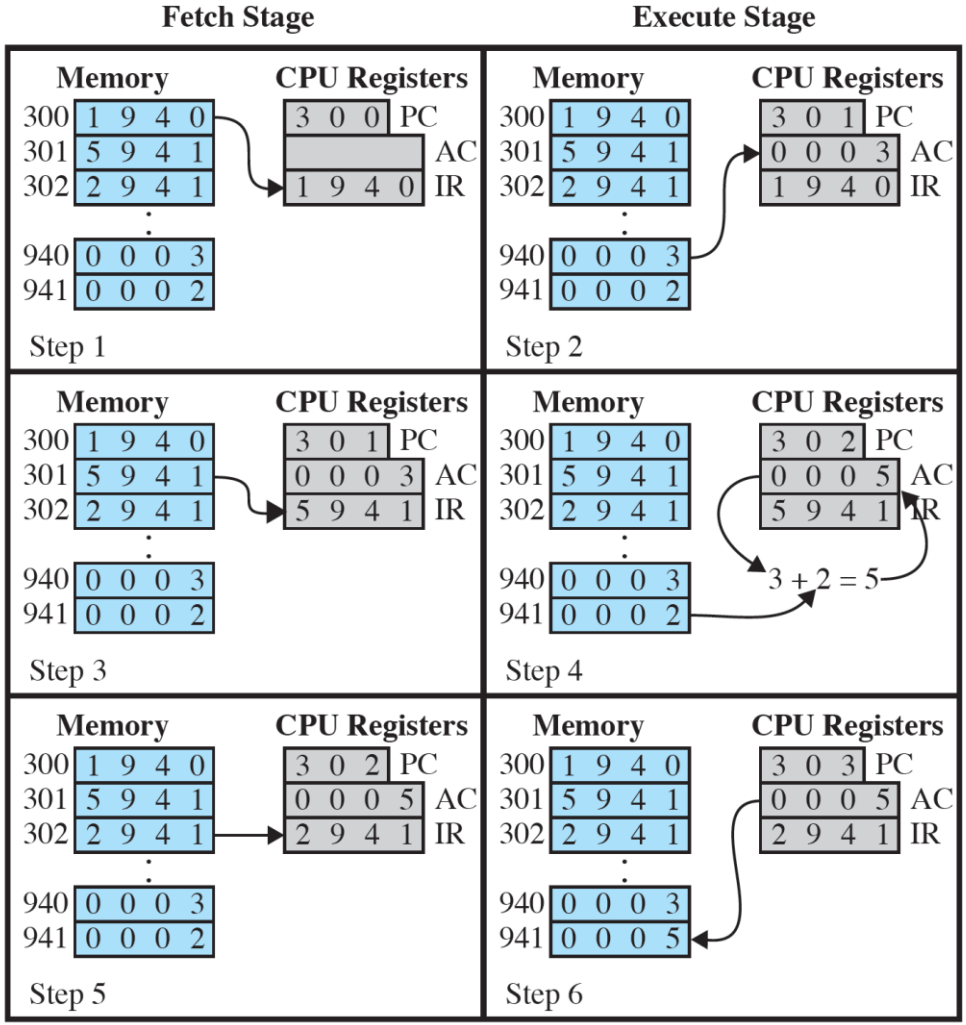


Elementy systemu komputerowego



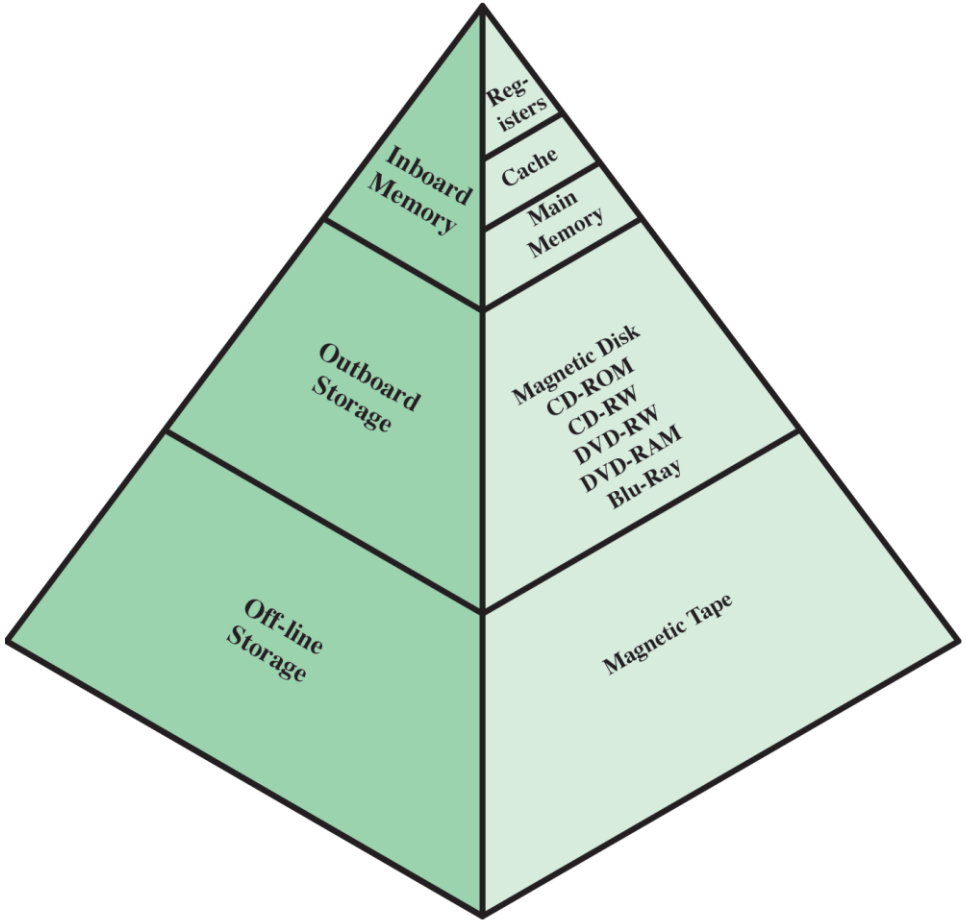
- PC – licznik rozkazów
- IR – rejestr instrukcji
- MAR – specyfikuje adresy w pamięci dla następnej operacji odczytu lub zapisu,
- MBR – zawiera dane zapisywane do pamięci lub odczytywane z pamięci
- I/OAR – określ konkretne urządzenie We/Wy
- I/OBR – używane do wymiany danych między modułem We/Wy i procesorem

Przykładowe wykonanie programu komputerowego



- Wykonanie programu, dodanie zawartości adresu 940 do zawartości adresu 941

Hierarchia pamięci



- Wraz z przemieszczaniem się w dół hierarchii
  - Maleje koszt każdego bitu
  - Wzrasta pojemność
  - Wzrasta czas dostępu
  - Wzrasta częstość dostępu do pamięci przez procesor

Elementy składowe systemu

- **zarządzanie procesami** - tworzenie i usuwanie procesów użytkowych i systemowych, wstrzymywanie i wznawianie procesów, mechanizmy synchronizacji procesów, mechanizmy komunikacji między procesami, mechanizmy obsługi zakleszczeń
- **zarządzanie pamięcią operacyjną** - utrzymywanie ewidencji aktualnie zajętych części pamięci, wybór procesów ładowanych do wolnych obszarów pamięci, stosowne do potrzeb przydzielanie i zwalnianie obszarów pamięci
- **zarządzanie plikami** - tworzenie i usuwanie plików i katalogów, dostarczanie operacji do manipulowania plikami i katalogami, odwzorowywanie plików na obszary pamięci pomocniczej, składowanie plików na stałych nośnikach pamięci
- **zarządzanie systemem WE/WY** - obsługa buforowania, pamięci podręcznej i spoolingu, obsługa ogólnego interfejsu do modułów sterujących urządzeń, obsługa modułów sterujących poszczególnych urządzeń
- **zarządzanie pamięcią pomocniczą** -zarządzanie obszarami wolnymi, przydzielanie pamięci, planowanie przydziału obszarów pamięci dyskowej

Elementy składowe systemu 2

- **praca sieciowa -** dostęp do sieci, zasobów dzielonych
- **system ochrony -** nadzorowanie dostępu do programów, procesów i plików
- **system interpretacji poleceń -** tworzenie procesów, zarządzanie procesami, obsługa WE/WY, administrowanie pamięcią operacyjną i pomocniczą, dostęp do plików, ochrony, sieci

Usługi systemu operacyjnego (1)

- **Usługi dla użytkownika:**
  - **wykonanie programu -** załadowanie do pamięci, uruchomienie, zakończenie
  - **operacje WE/WY -** na pliku lub urządzeniu
  - **manipulowanie systemem plików -** tworzenie, usuwanie, zapisywanie, odczytywanie plików
  - **komunikacja -** pamięć wspólna i komunikaty
  - **wykrywanie błędów**

Usługi systemu operacyjnego (2)

- **Usługi do optymalizacji działania systemu:**
  - **przydzielanie zasobów**
  - **rozliczanie -** przechowywanie danych o tym, jak użytkownicy korzystają z zasobów systemu - do wystawiania rachunków lub celów statystycznych
  - **ochrona -** nadzór nad dostępami do zasobów systemu, zabezpieczenie systemu przed niepożądanymi czynnikami zewnętrznymi - użytkownik musi uwierzytelnić w systemie swoją tożsamość