

# Podstawy systemów operacyjnych

dr inż. Beata Marciniak



# Literatura

- A. Silberschatz, J.L. Peterson, P.B. Galvin  
Podstawy systemów operacyjnych.
- A.M. Lister, R.D. Eager  
Wprowadzenie do systemów operacyjnych.
- W. Stallings  
Systemy operacyjne. Struktura i zasady budowy

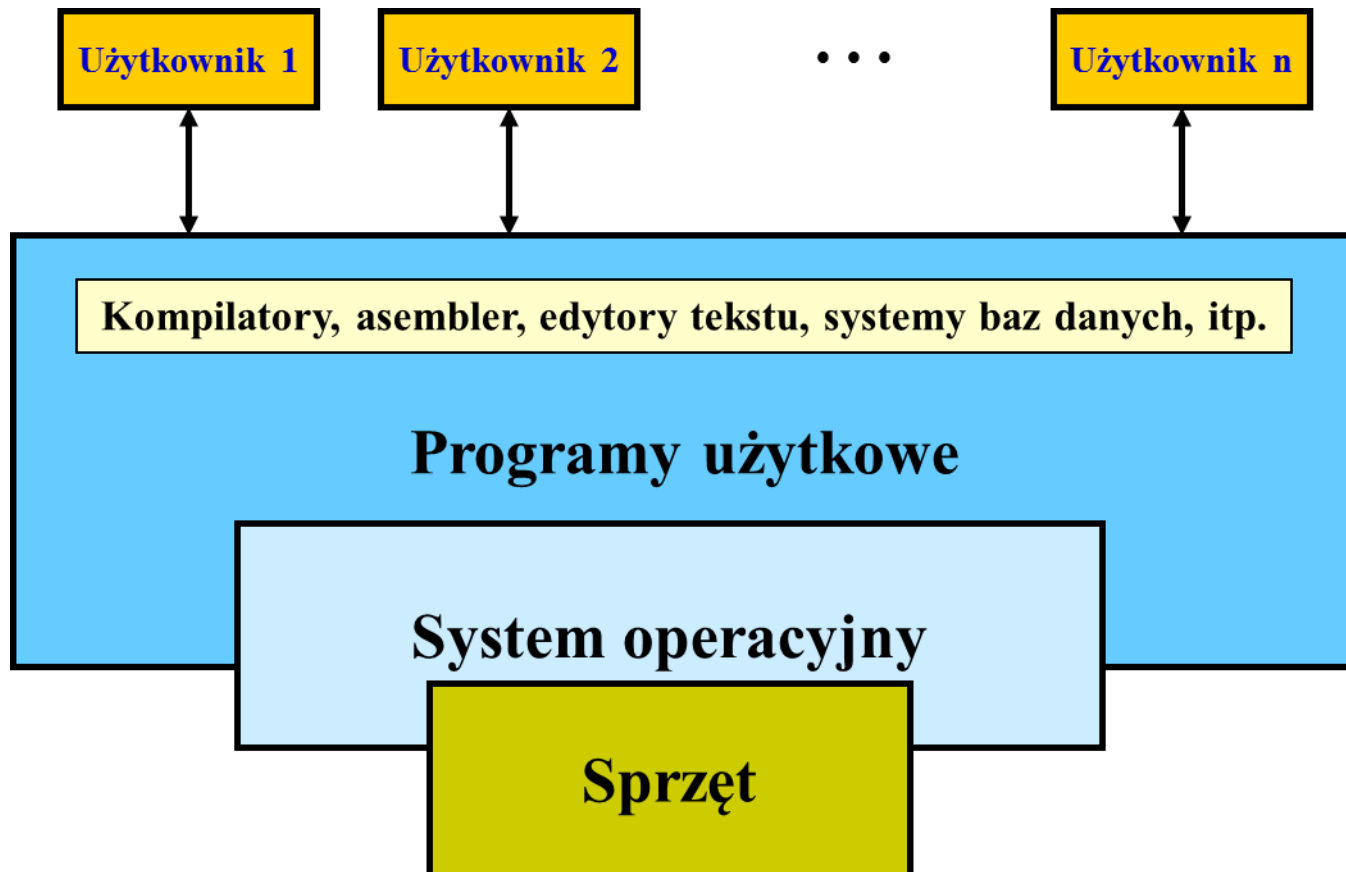
# Zaliczenie

- Wykład 30 godzin
- Laboratorium 30 godzin
- Zaliczenie wykładu w postaci testu elektronicznego (forms) – część pytań zamkniętych, część pytań otwartych.
- Zaliczenie:
  - 16 grudnia 2020 – I część
  - 20 stycznia – II część
  - 27 stycznia – cały materiał
  - sesja poprawkowa – ostatni termin

# Definicja systemu operacyjnego

- Systemem operacyjnym nazywamy specjalny program komputerowy, który **steruje i zarządza** wszystkimi zasobami systemu komputerowego (sprzęt i oprogramowanie) oraz **zapewnia użytkownikowi łatwy interfejs** dla dostępu do wirtualnej maszyny (wirtualnego komputera).
- System operacyjny jest programem, który działa jako pośrednik pomiędzy użytkownikiem a sprzętem komputerowym.

# System komputerowy

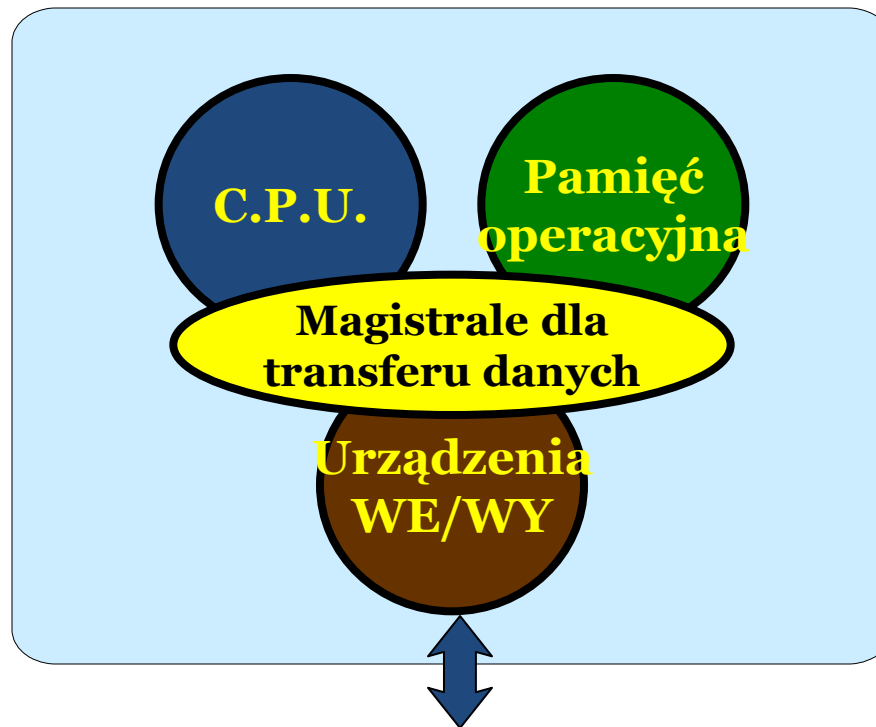


# Perspektywa systemu

- Z perspektywy systemu komputerowego, system operacyjny jest programem najbliższym powiązanym ze sprzętem (dystributor zasobów).
- Istnienie systemów operacyjnych jest uzasadnione tym, że umożliwiają one rozsądne rozwiązywanie problemów np. z dostępem do urządzeń wejścia – wyjścia, wspólnym sterowaniem zasobami czy gromadzeniem danych.

# Sprzęt systemu komputerowego

1. Procesor (jednostka centralna CPU);
2. Pamięć operacyjna (program i dane);
3. Urządzenia WE/WY.



# Zadania systemu operacyjnego

## Komputer PC

- Udostępnienie użytkownikom wszystkich możliwości systemu komputerowego w sposób łatwy i wygodny.

## Serwer, komputer główny

- Koordynacja pracy elementów systemu komputerowego w celu efektywnego oraz wydajnego wykorzystywania systemu (np. dla realizacji systemów wielodostępowych).



# Zadania systemu operacyjnego

- Ukrywa szczegóły sprzętowe systemu komputerowego poprzez tworzenie abstrakcji (maszyn wirtualnych).
- Przykłady:
  - jednolity sposób dostępu do urządzeń zewnętrznych
  - zbiory bloków dyskowych widziane jako pliki o symbolicznych nazwach
  - duża, szybka, dedykowana pamięć operacyjna - współbieżne wykonanie programów (jako abstrakcja równoległości)

# Zadania systemu operacyjnego c.d.

- Zarządza zasobami:
  - zasoby to obiekty niezbędne do wykonania programu, np. pamięć, czas CPU, wejście-wyjście, porty komunikacyjne
  - strategie przydziału i odzyskiwania zasobów (zarządzanie pamięcią, zarządzanie procesorem, zarządzanie plikami, zarządzanie urządzeniami)
  - efektywność zarządzania zasobami decyduje o wydajnej eksploatacji sprzętu komputerowego
- Dostarcza przyjazny interfejs - wygoda użycia (ustawianie przełączników, karty perforowane, taśmy perforowane, terminale graficzne z myszką i klawiaturą).



# Historia rozwoju systemów komputerowych

# Wczesne systemy operacyjne

1. System komputerowy, to przede wszystkim sprzęt.
2. Do pracy z komputerem wymagane były *wysokie* kwalifikacje.
3. Wczesne oprogramowanie - niezbyt przyjazny interfejs (asembler, programy ładujące, programy łączące, biblioteki typowych funkcji, kompilatory, programy sterujące urządzeń).
4. Programista **tworzył, uruchamiał, testował** oprogramowanie oraz nadzorował osobiście **interakcyjny** proces obliczeniowy.
5. Bezpośredni, sposób pracy. Dostęp do komputera zgodnie z ustalonym wcześniej harmonogramem.
6. Pierwsze języki wysokiego poziomu Fortran, Cobol, Algol (kompilatory oraz konieczność ich instalowania).

# Proste systemy wsadowe

- Zatrudnienie *operatora* (użytkownik <> operator)
- Skrócenie czasu instalowania zadania przez przygotowywanie **wsadu** zadań o podobnych wymaganiach sprzętowych
- Automatyczne porządkowanie zadań - automatyczne przekazywanie sterowania od jednego zadania do drugiego
- Rezydentny program **monitor**
  - inicjalnie sterowanie jest w monitorze
  - przekazanie sterowania do zadania
  - po zakończeniu zadania sterowanie wraca do monitora
- Wprowadzenie kart sterujących (Job Control Language)
- Istotna zmiana trybu pracy z punktu widzenia użytkownika
- Zwiększona przepustowość systemu kosztem średniego czasu przetworzenia zadania

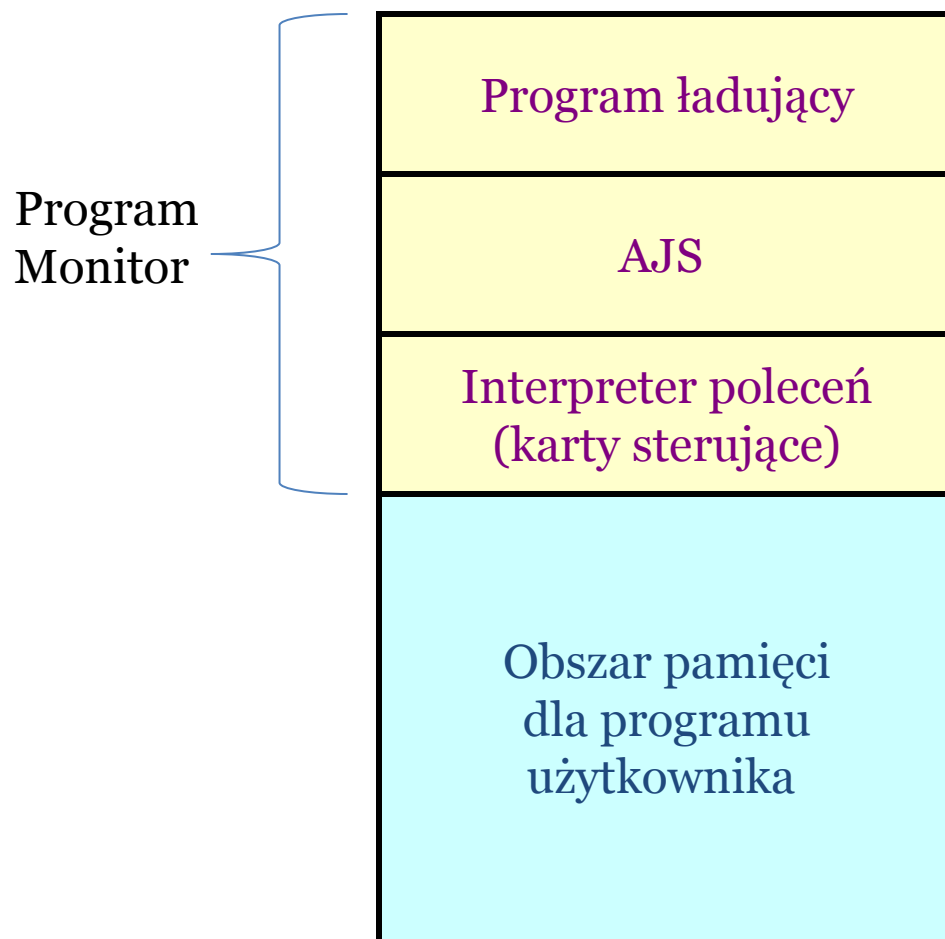
**Problem:** niska wydajność (CPU i urządzenia wejścia-wyjścia nie mogą pracować równocześnie, czytnik kart bardzo wolny)

**Rozwiązanie:** praca w trybie pośrednim (off-line)

# Program Monitor

1. Efektywne wykorzystanie drogiego sprzętu wymusiło zatrudnienie wyedukowanych operatorów sprzętu.
2. Operator realizował zadania zgodnie z porządkiem niezależnie od programistów.
3. Powstał wsadowy tryb pracy systemu (ten sam typ zadań realizowany razem, bez rekonfiguracji systemu).
4. Pojawia się koncepcja automatycznego porządkowania zadań (**Automatic Job Sequences**) oraz prosty program zarządzający, rezydujący w pamięci komputera – Monitor.

# Pamięć z systemem Monitor



**AJS** – Automatic Job Sequences

**JCL** – Job Control Language

Program Monitor przekazuje sterowanie do poszczególnych zadań, które po zakończeniu zwracają do niego sterowanie. (Automatyczny Operator )

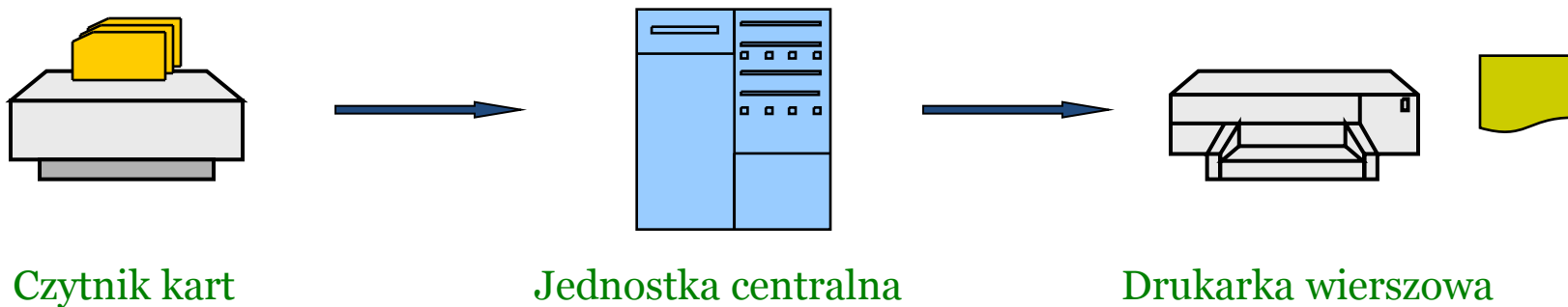
# Praca pośrednia

1. Wolne urządzenia wejściowe (np. czytniki kart) pracowały nieefektywnie z coraz szybszymi procesorami.
2. Szybkie urządzenia wejściowe (taśmy magnetyczne) mogły pracować w trybie WE oraz WY oraz współpracować z czytnikami kart oraz drukarkami.
3. Pojawiły się proste systemy komputerowe o ograniczonych możliwościach (przetwarzanie satelitarne).
4. Praca w trybie pośrednim pozwala tworzyć programy z logicznymi urządzeniami WE-WY (niezależne od urządzeń fizycznych).

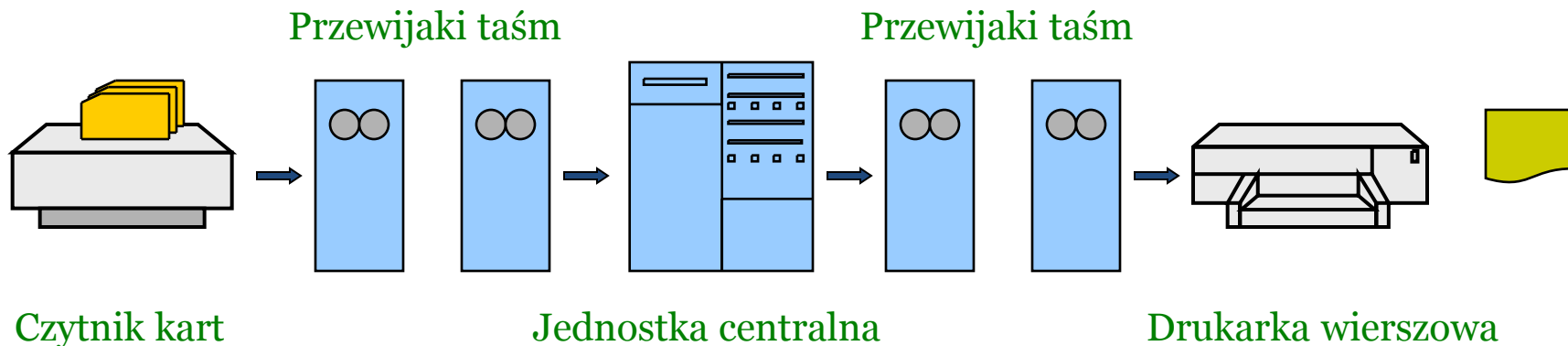


# Tryby pracy

## Praca bezpośrednia



## Praca pośrednia



# Buforowanie

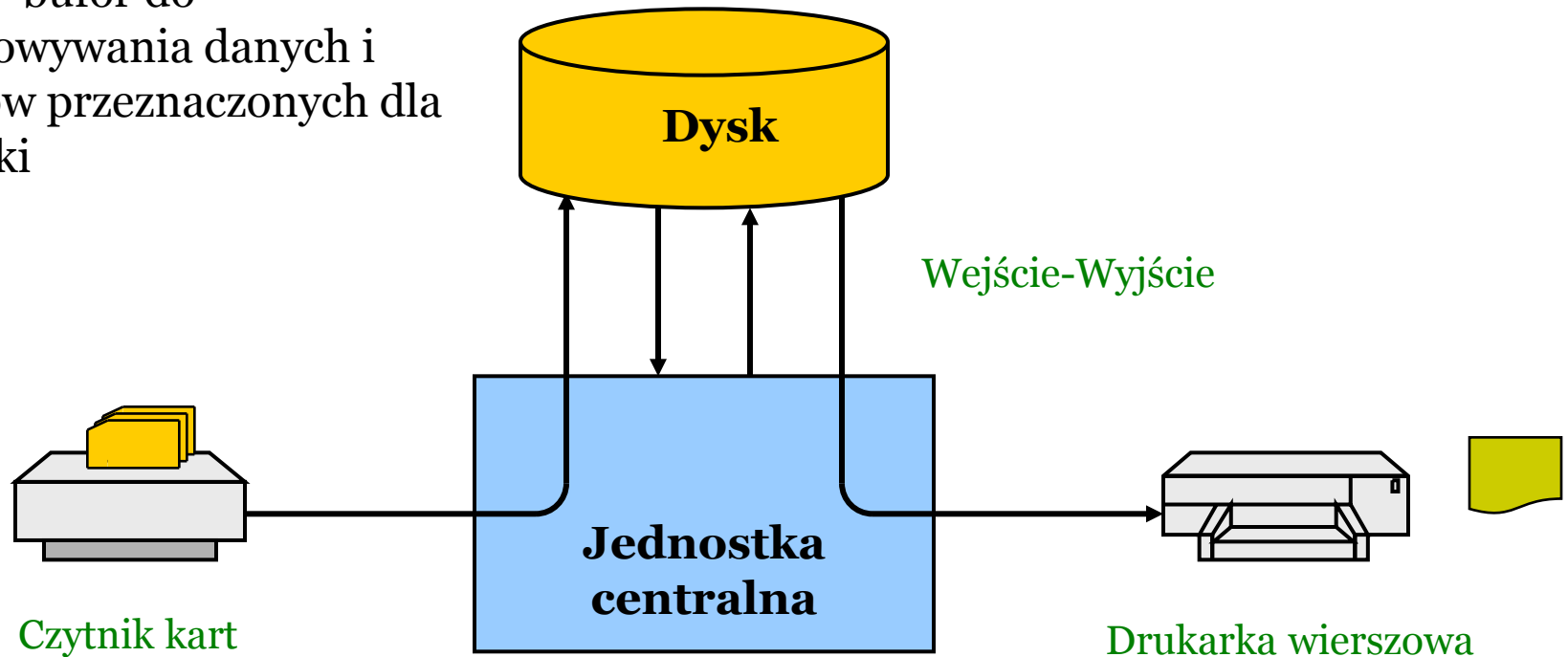
1. Pomędzy urządzeniami WE, WY i procesorem znajduje się specjalny moduł do przechowywania danych – bufor. Urządzenie WE pisze dane do bufora (WY czyta), zaś procesor czyta dane (pisze dane). Bufor może przechowywać kilka danych.
2. Z reguły jedno z urządzeń (WE, WY lub procesor) pracuje szybciej od drugiego i wtedy:
  - procesor czeka na urządzenie WE (WY) – zadanie uzależnione od WE-WY;
  - urządzenia WE-WY czekają na procesor – zadanie uzależnione od jednostki centralnej.

# Spooling

1. Pojawienie się pamięci dyskowych pozwoliło usprawnić pracę pośrednią i buforowanie (łatwiejszy dostęp do danych w stosunku do taśm – **dysk stał się buforem**).
2. Możliwe stało się gromadzenie danych w różnych miejscach na dysku (w taśmach dane musiały być zapisywane sekwencyjnie).
3. Jednostka centralna uzyskała łatwy dostęp do danych (zarówno dla WE jak i WY).
4. Spooling (**Simultaneous Peripheral Operation On-Line**) daje możliwość buforowania zadań dla urządzeń nie tolerujących przeplatania danych (**np. dla drukarki**) – kolejka zadań do wykonania.

# Spooling

Spool – bufor do przechowywania danych i wyników przeznaczonych dla drukarki



# Wieloprogramowość



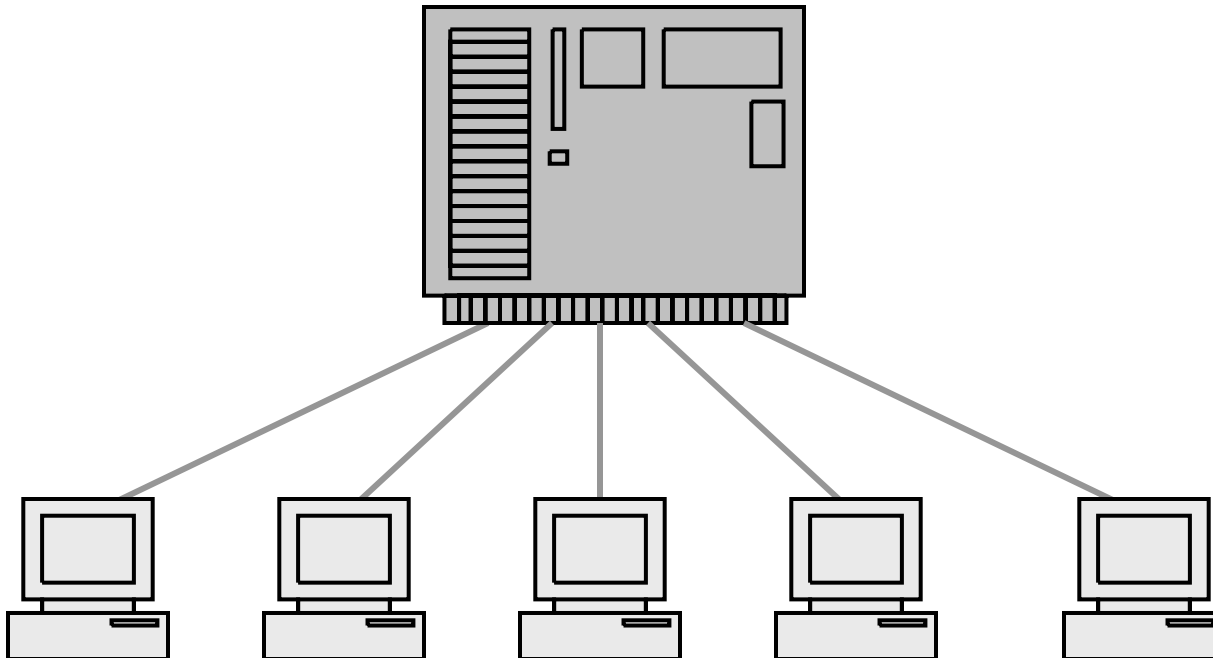
Komputer wykonuje *i-te* zadanie (aktywny proces obliczeniowy) umieszczone w pamięci.  
Gdy *i-ty* proces musi oczekiwać na jakiś zasób, wówczas procesor zawiesza ten proces i przechodzi do wykonania kolejnego nieaktywnego procesu.

**Mapa pamięci komputera**

# Wielozadaniowość

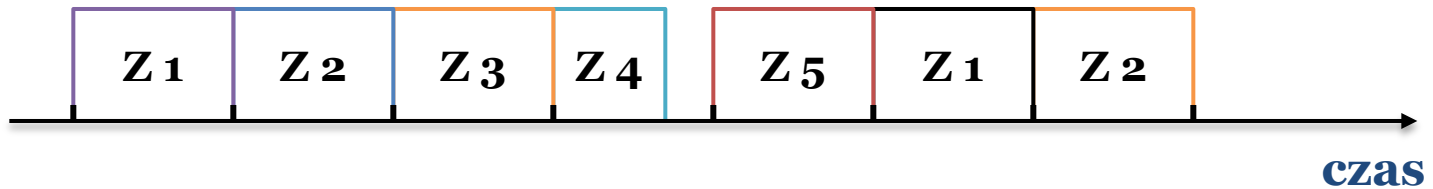
**Wielozadaniowość.** Przełączanie między aktywnymi zadaniami – podział czasu.

**Wymaga systemu plików dostępnych bezpośrednio.**

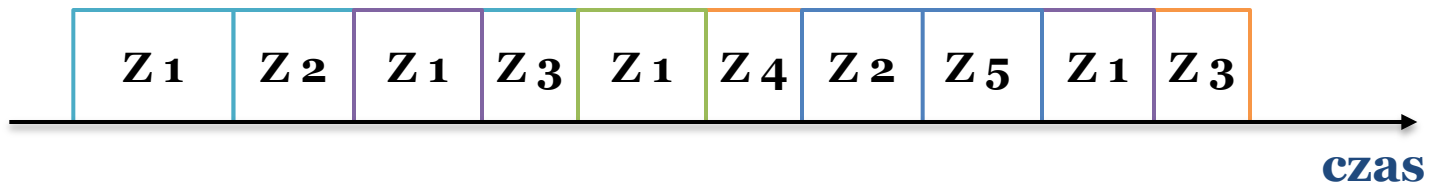


# Wielozadaniowość

System wielozadaniowy ze stałym podziałem czasu (R-R)

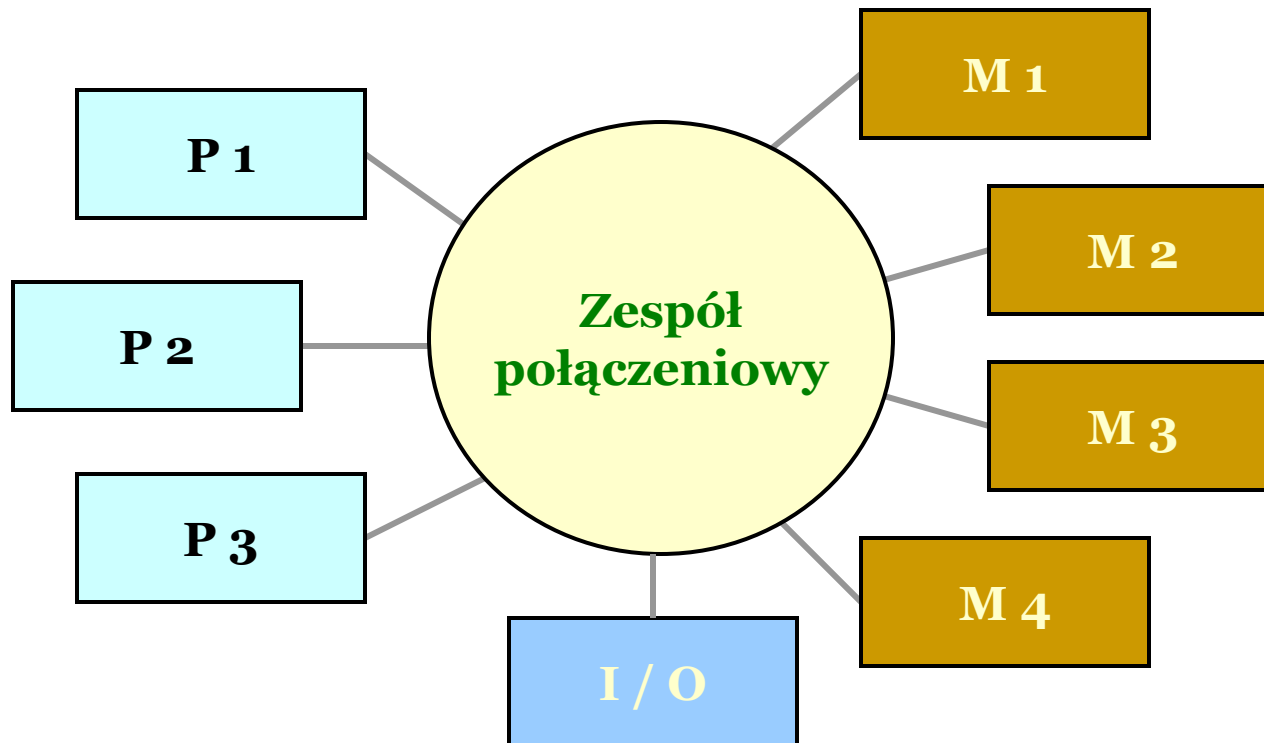


System wielozadaniowy reagujący na zdarzenia (RT)



# Systemy rozproszone

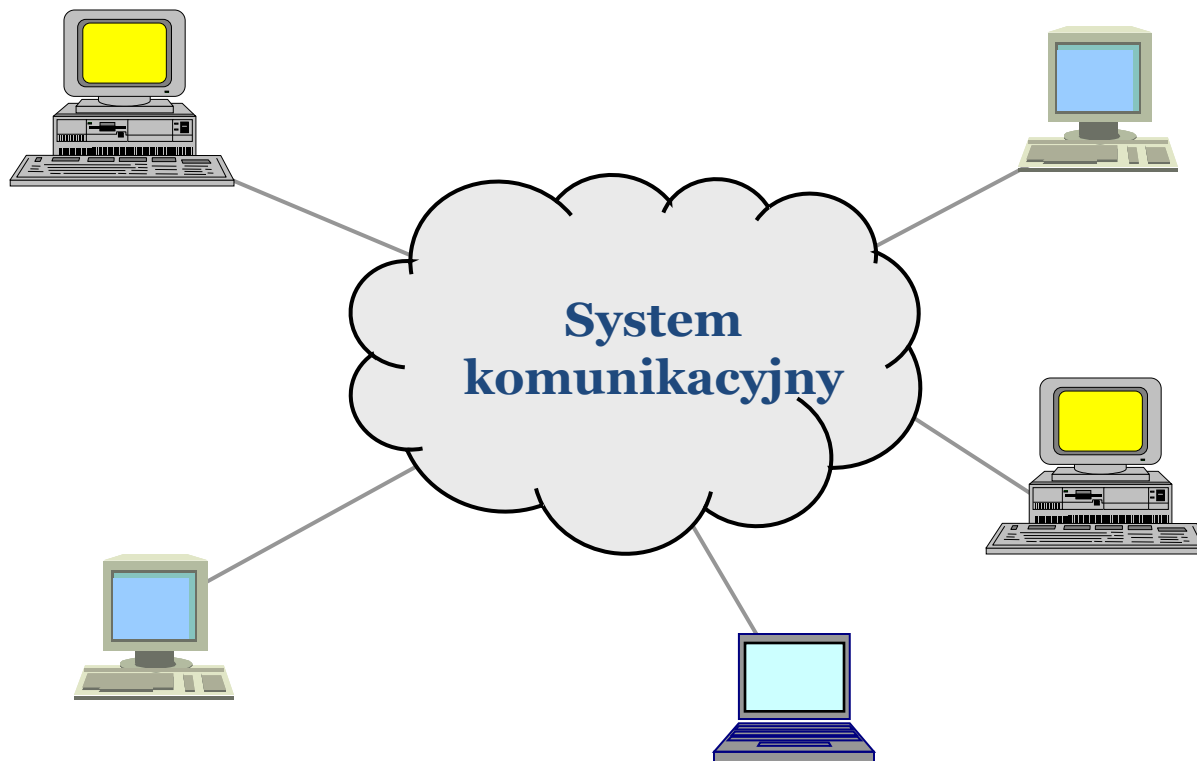
Systemy wieloprocessorowe: ściśle powiązane – ciasno sprzężone.





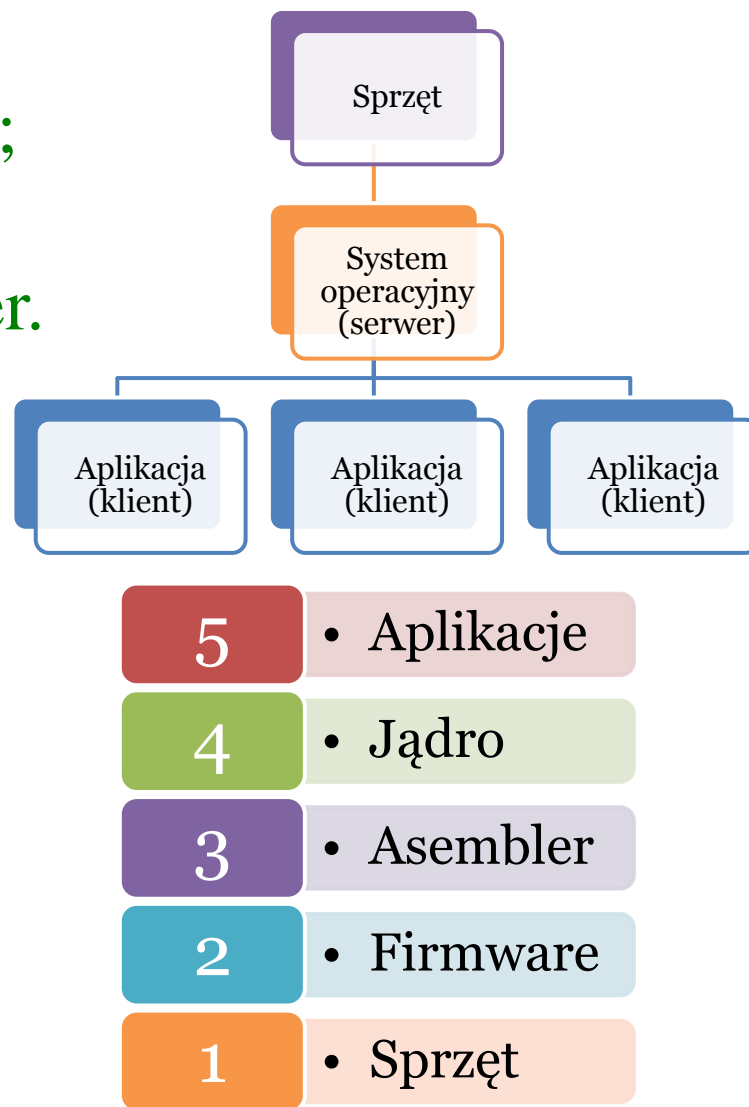
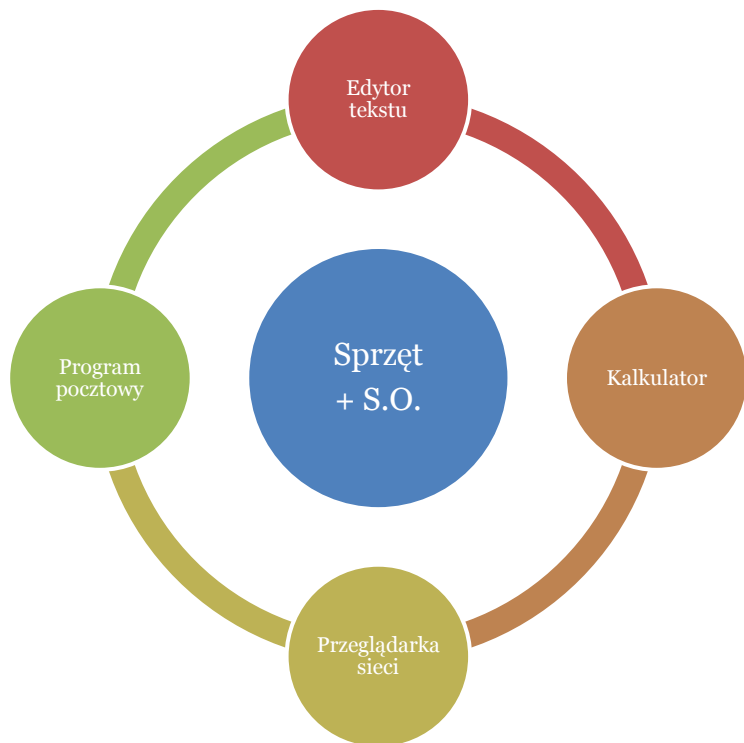
# Systemy rozproszone

Systemy wielokomputerowe: luźno sprzężone – sieci komputerowe.



# Struktury systemów operacyjnych

1. struktura jednolita (prosta);
2. struktura warstwowa;
3. struktura typu klient-serwer.



# Rodzaje systemów operacyjnych

1. systemy dla indywidualnego użytkownika (np. DOS, Windows, Linux), również z możliwością wielodostępu (np. Windows Serwer, Unix);
2. systemy operacyjne dla sterowania procesami (systemy czasu rzeczywistego, np. RTOS, QNX);
3. systemy dostępu do dużych baz danych – systemy przeszukiwania plików (np. system klasy SQL);
4. systemy dla przetwarzania transakcji (np. systemy rezerwacji miejsc, systemy obsługi giełdy).

Systemy operacyjne pracujące w chmurze



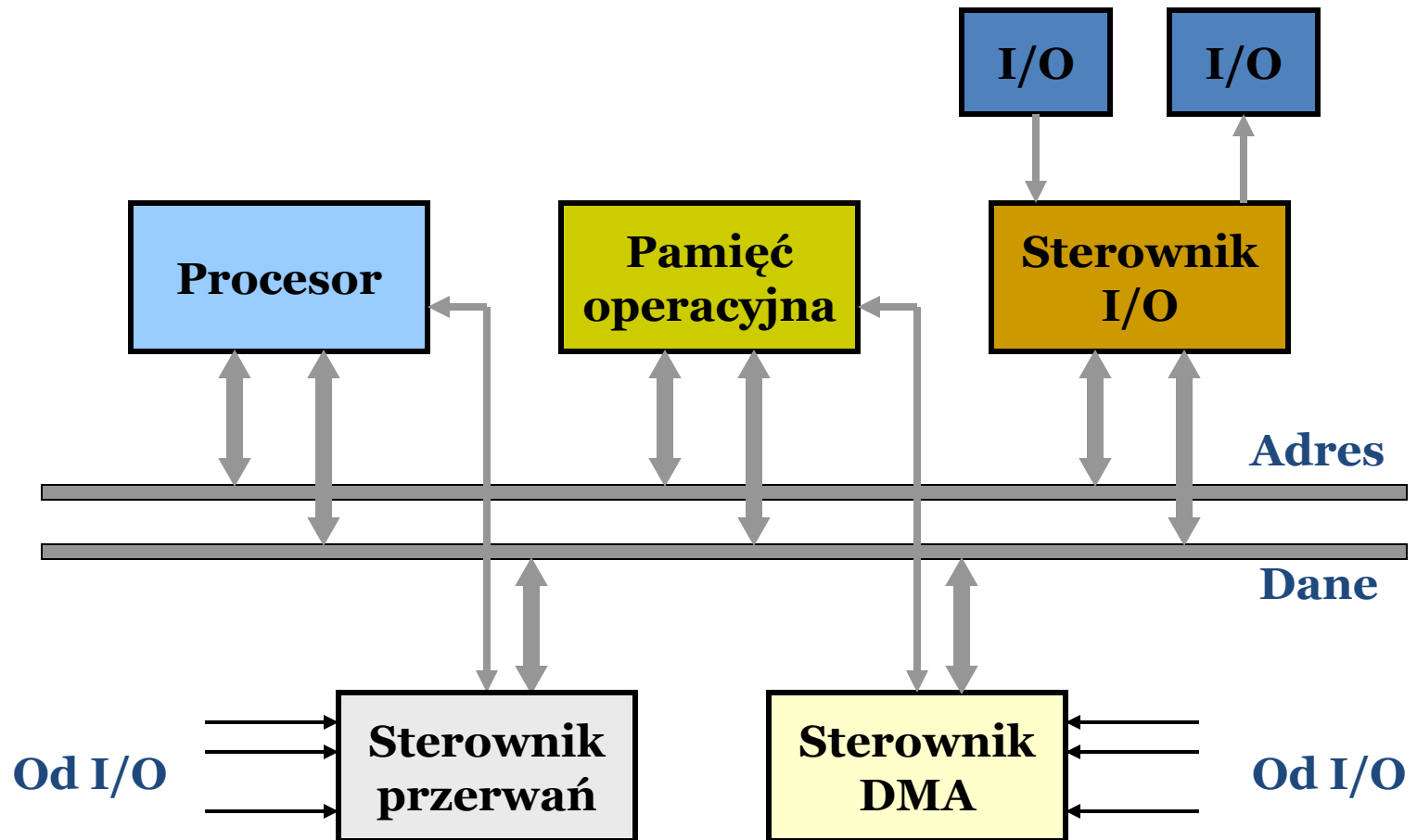
# Pierwsze systemy komputerowe

# Mechanizmy komunikacji

Istnieją trzy mechanizmy realizacji operacji we/wy:

- Programowe we/wy;
- Układ we/wy sterowany przerwaniami;
- Bezpośredni dostęp do pamięci (DMA).

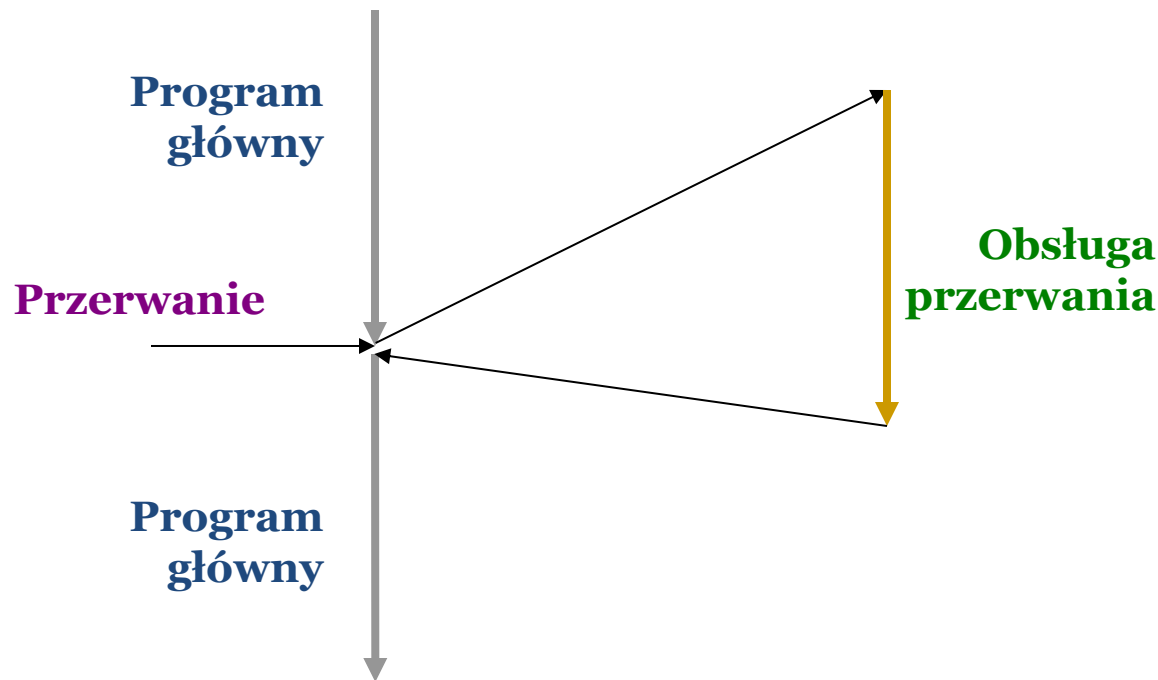
# Architektura systemu



# Programowe we/wy

1. Kiedy procesor wykonuje program lub napotyka rozkaz odnoszący się do operacji we/wy, realizuje taki rozkaz przez wysłanie polecenia do odpowiedniego modułu we/wy.
2. W przypadku programowego we/wy, moduł we/wy wykonuje żadaną czynność i ustawia odpowiednie bity w rejestrze stanu modułu we/wy.
3. Zestawy rozkazów obejmują rozkazy we/wy podzielone na kategorie:
  - a) **Sterowanie:** aktywowanie zewnętrznego urządzenia i sterowania nim (np. taśma ustawia się na właściwym rekordzie).
  - b) **Status:** służy do testowania różnorodnych statusów skojarzonych z modułem we/wy oraz jego urządzeniami peryferyjnymi.
  - c) **Transfer:** wykorzystywany do odczytu i/lub zapisu pomiędzy procesorem a rejestrami oraz urządzeniami zewnętrznymi.

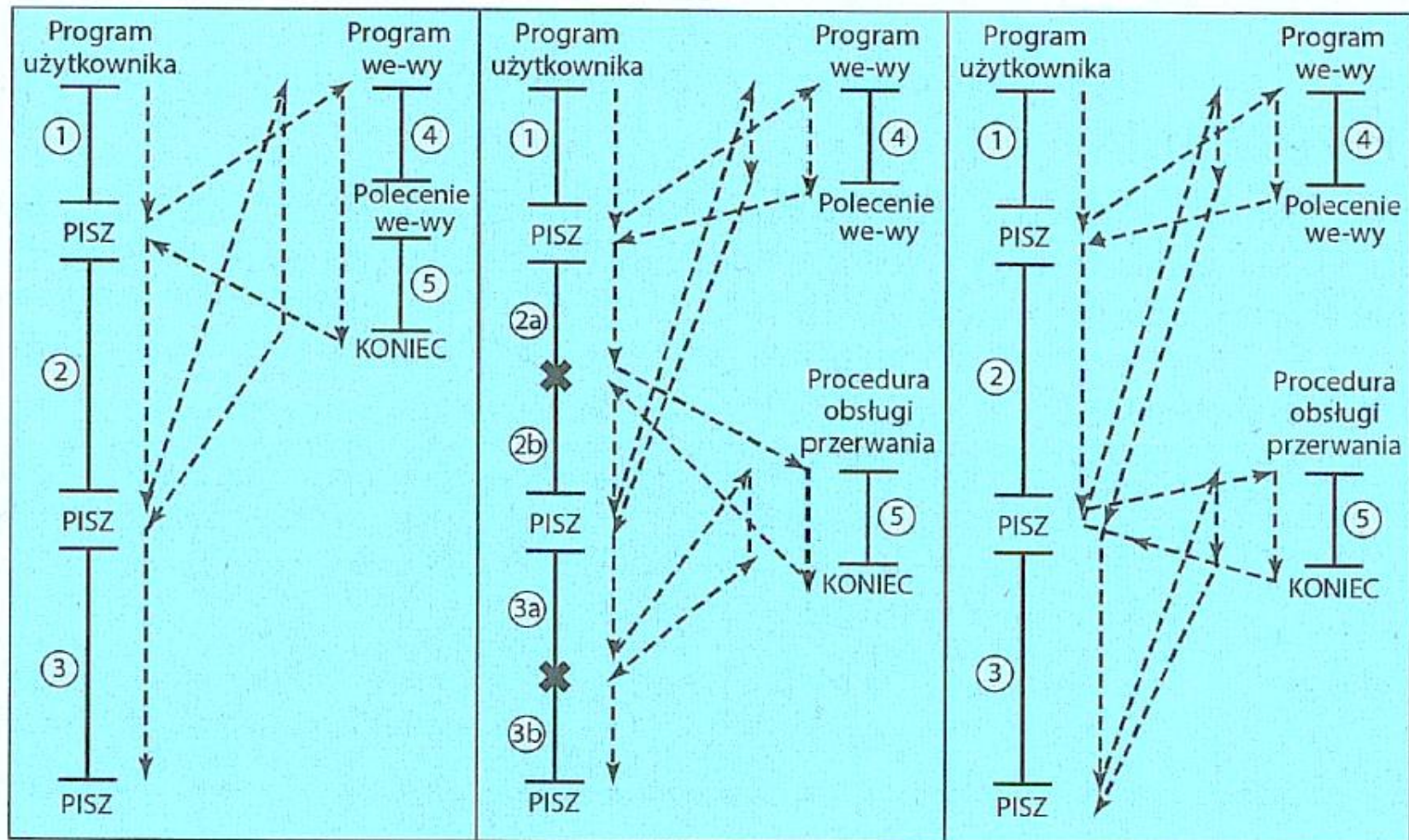
# Przerwanie w systemie



## **Układ we/wy sterowany przerwaniem**

Procesor wyśtosowuje polecenie np. zapisu danych. Następnie zapisuje kontekst bieżącego i rozpoczyna inne zadanie. Na koniec każdego cyklu rozkazów procesor sprawdza czy nie pojawiły się przerwania.





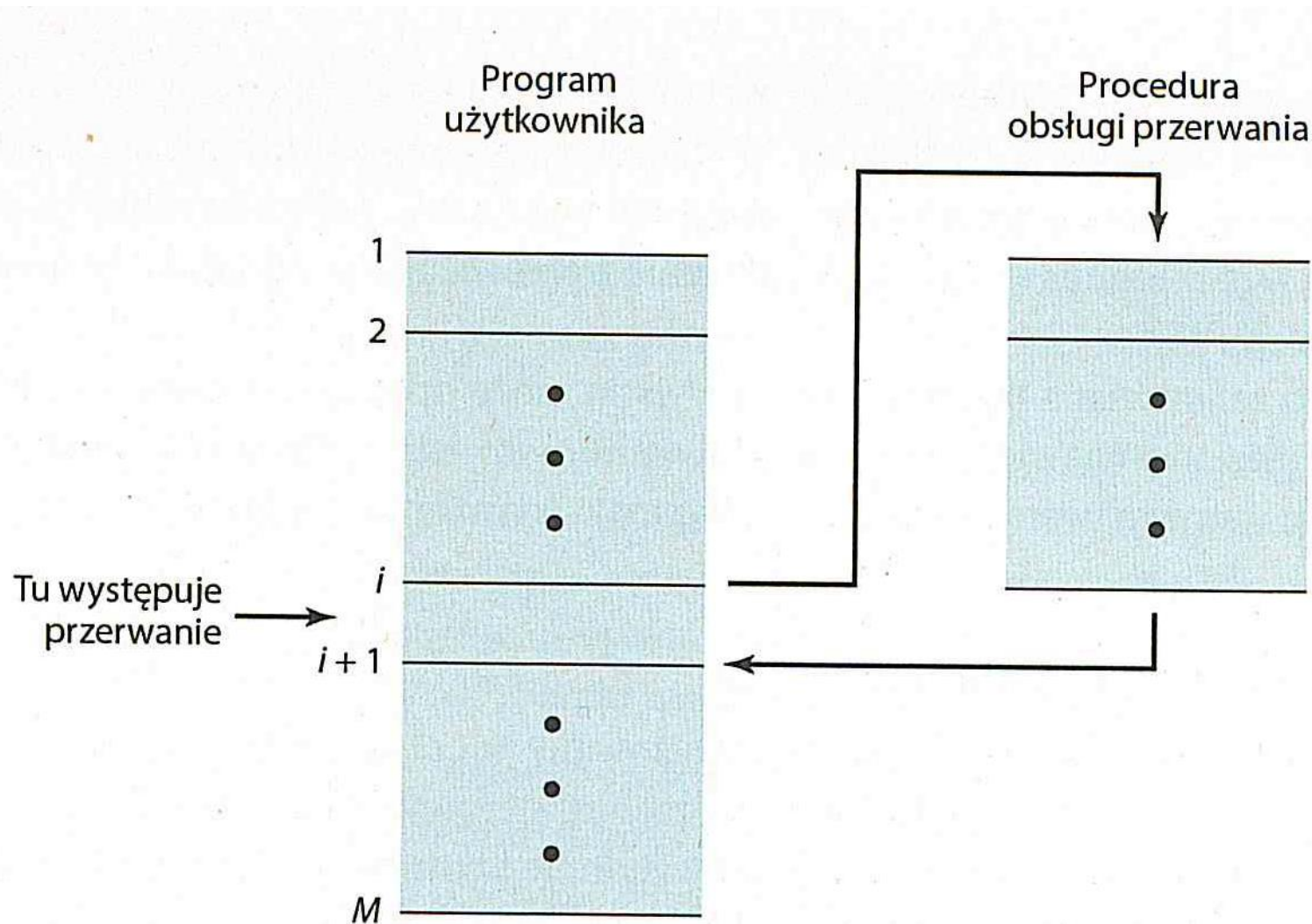
(a) Bez przerwania

(b) Przerwania;  
krótkie czekanie na we-wy

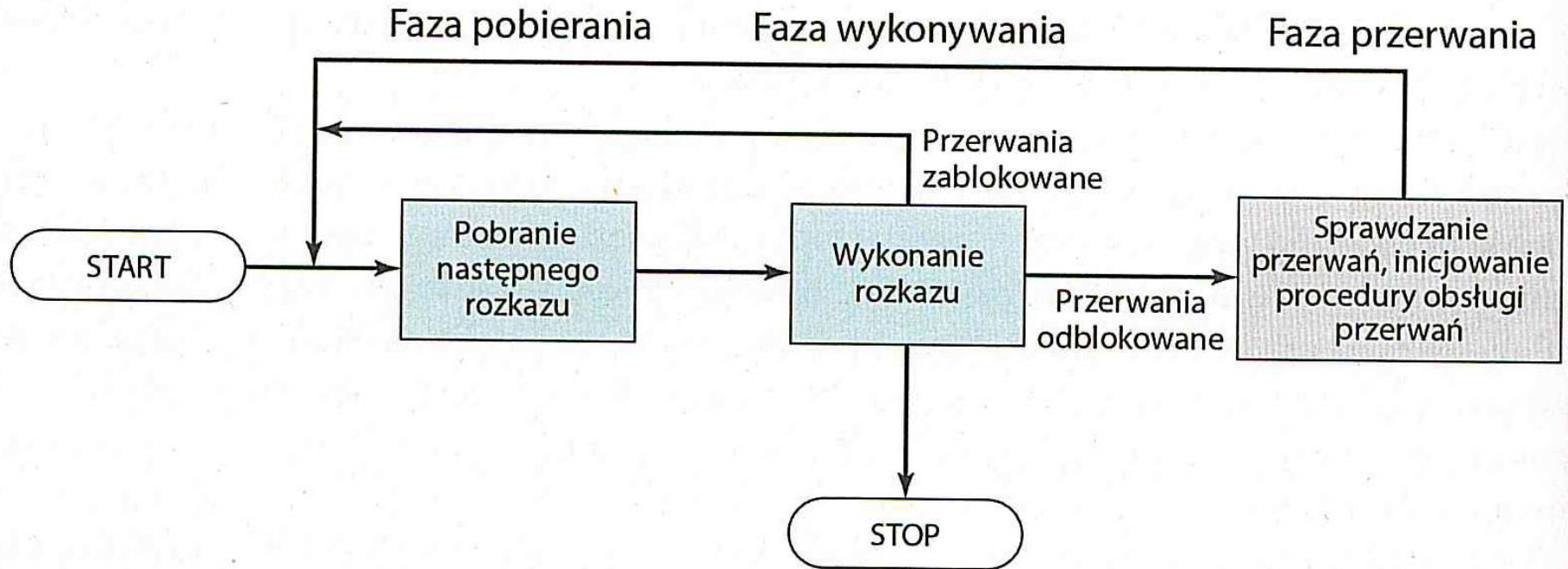
(c) Przerwania;  
długie czekanie na we-wy

✕ = przerwanie występuje w czasie wykonywania programu użytkownika

# Przekazanie sterowania

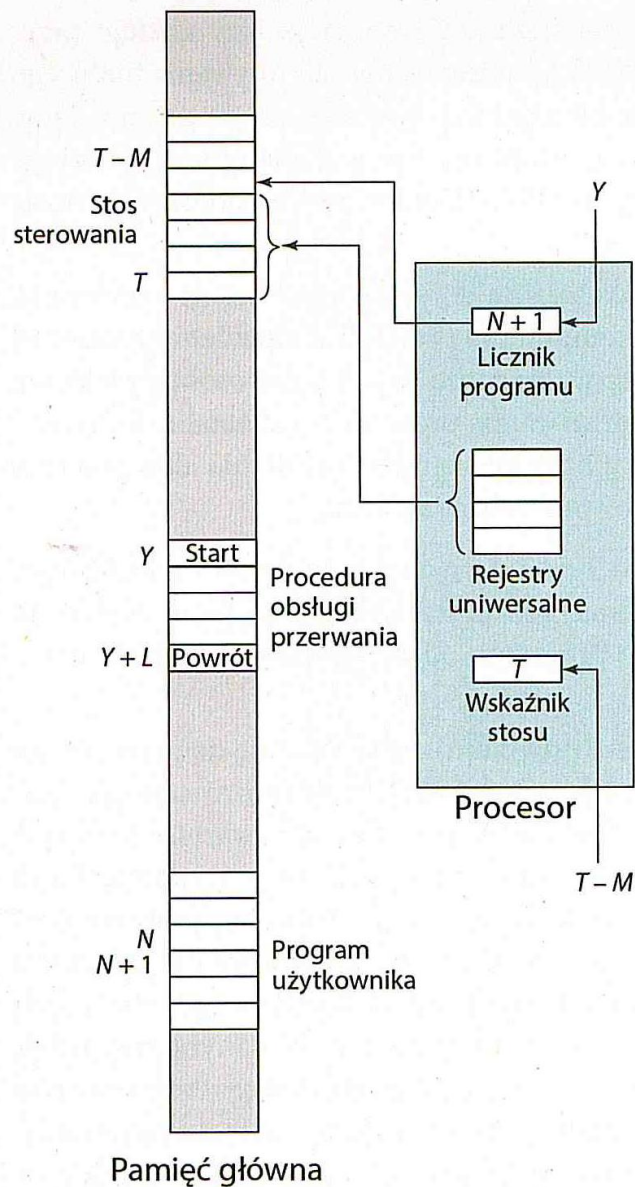


# Cykl rozkazowy z przerwaniami

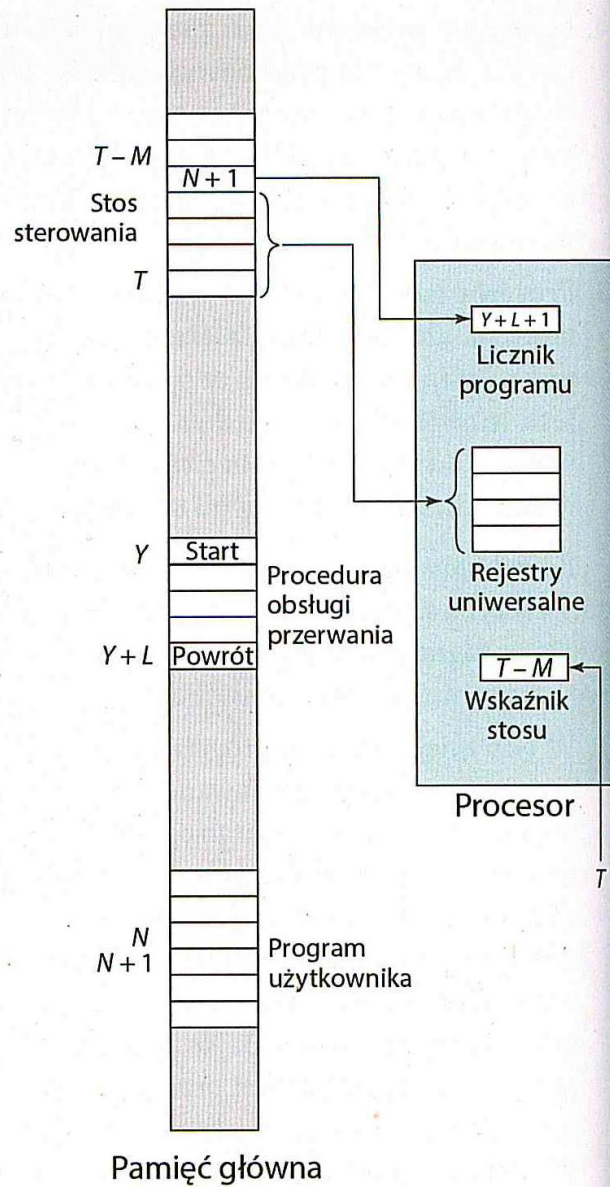




# Zmiany w pamięci i rejestrach

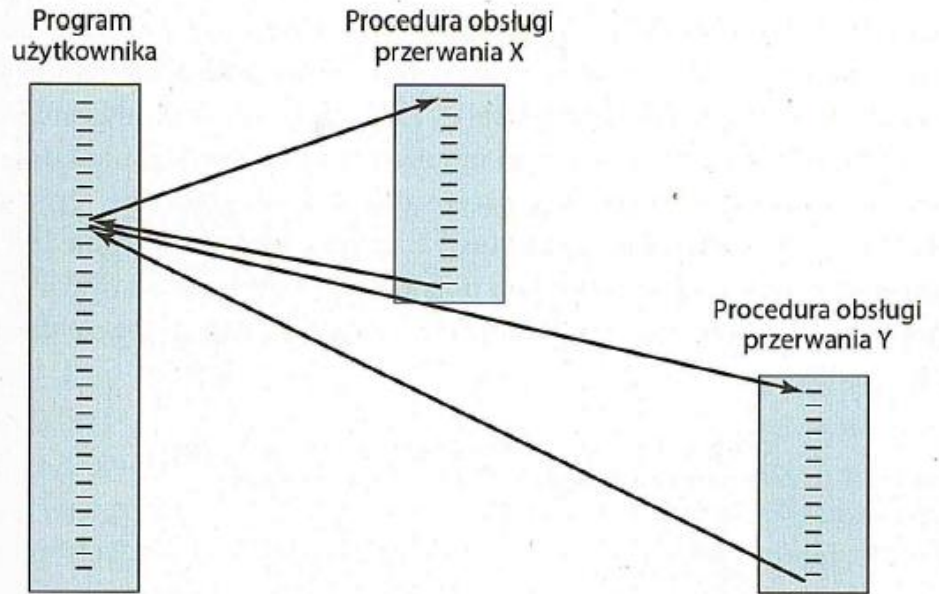


(a) Przerwanie występuje po rozkazie z komórki  $N$

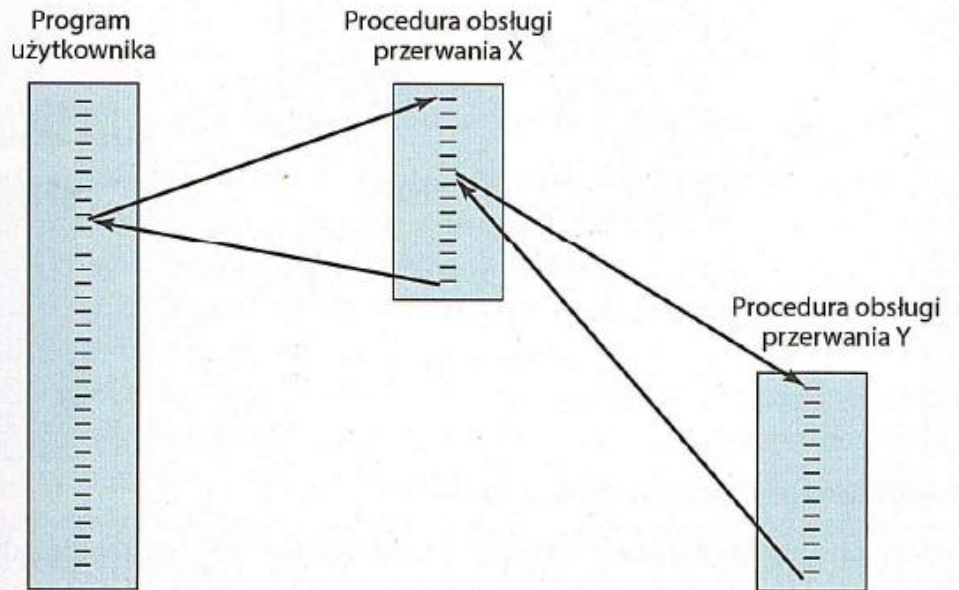


(b) Powrót z przerwania

# Przekazywanie sterowania w sytuacji wielu przerwań

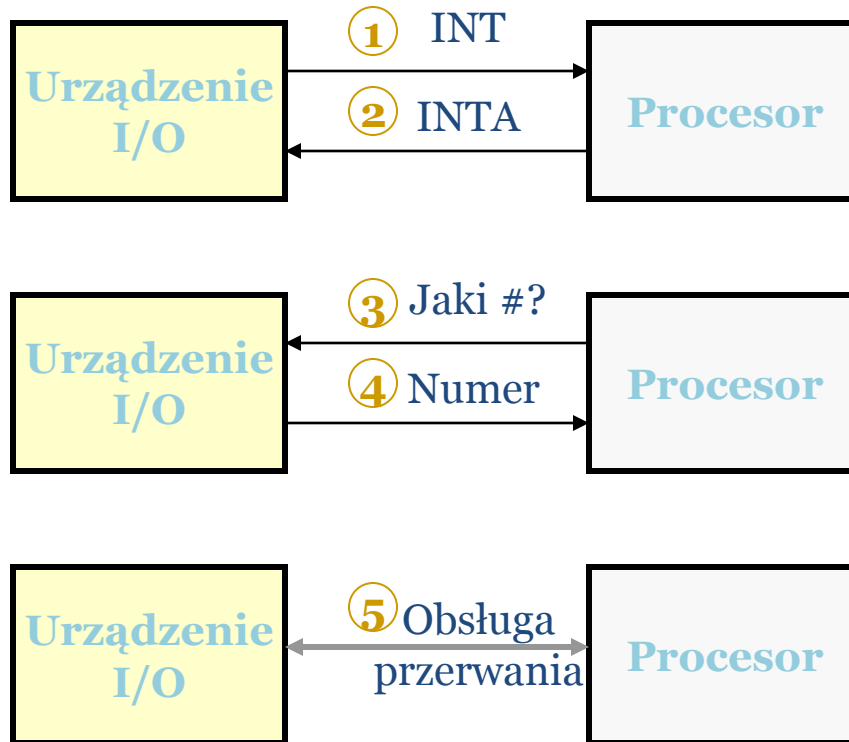


(a) Sekwencyjne przetwarzanie przerwań



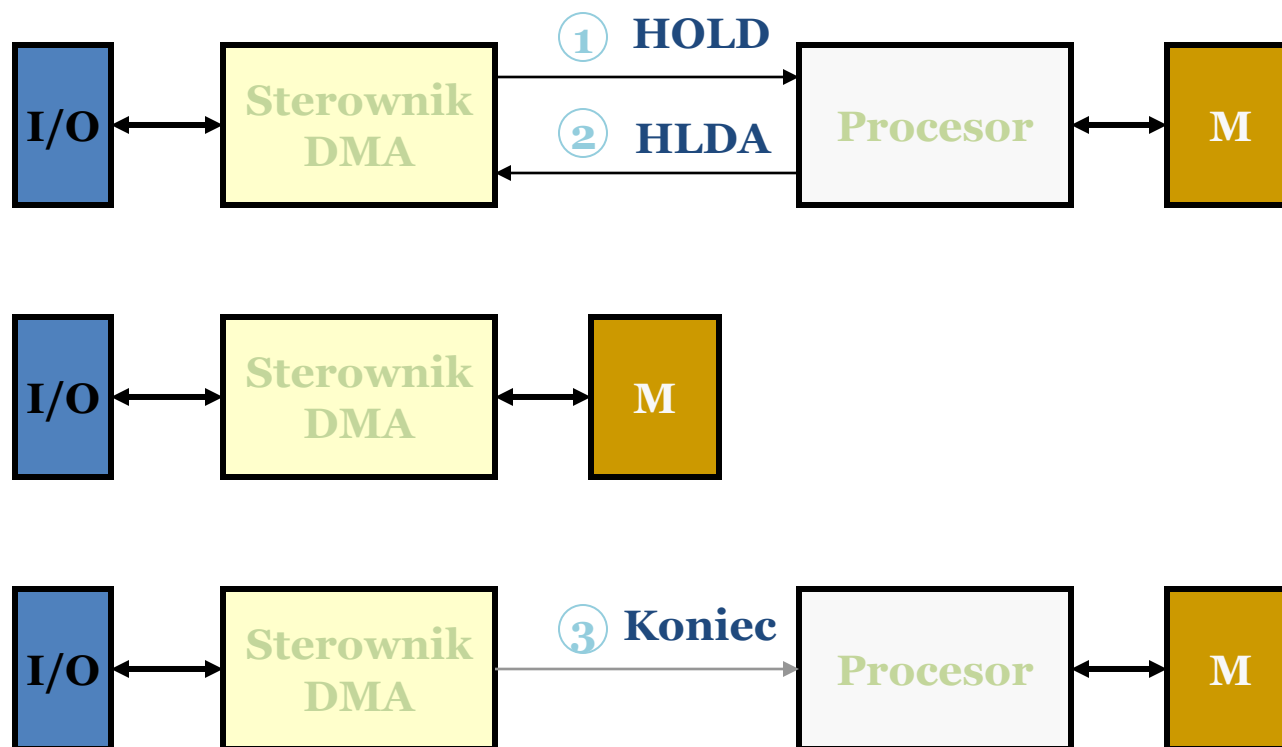
(b) Zagnieżdżone przetwarzanie przerwań

# Przerwanie w systemie



# Bezpośredni dostęp do pamięci (DMA)

ang. Direct Memory Access



# Charakterystyka DMA

W każdym przypadku, kiedy procesor chce odczytać lub zapisać blok danych, wysyła polecenie do modułu DMA z następującymi informacjami:

- rodzajem żądanego polecenia: odczyt/zapis,
- adresem urządzania we/wy,
- początkową lokalizacją w pamięci, gdzie zostanie przeprowadzona operacja zapisu/odczytu,
- liczba słów, które mają zostać zapisane/odczytane

W międzyczasie procesor będzie realizował inne zadania, operacje we/wy zostały przekazane do modułu DMA i ten moduł obsłuży tę operację.



# Urządzenia we/wy

- Podział typów urządzeń zewnętrznych:
  - urządzenia blokowe, możliwy odczyt/zapis każdego bloku niezależnie (dyski, nośniki danych),
  - urządzenia znakowe, łańcuch znaków bez podziału na bloki, nie ma adresowalności ani ustawianego wskaźnika bieżącej pozycji (drukarka, klawiatura),
  - czasem, ze względu na specyfikę, wyróżnia się jako osobną klasę urządzenia sieciowe/komunikacyjne,
  - niektóre urządzenia nie pasują do powyższej klasyfikacji, na przykład czasomierze (ang. timers),



# Architektura systemów operacyjnych

# Składowe moduły systemu

Moduły systemu operacyjnego służą do realizacji następujących funkcji:

1. funkcji zarządzania procesami;
2. funkcji zarządzania pamięcią operacyjną;
3. funkcji zarządzania pamięcią pomocniczą;
4. funkcji zarządzania systemem wejścia-wyjścia;
5. funkcji zarządzania plikami;
6. funkcji ochrony procesów w pamięci operacyjnej;
7. funkcji nadzoru pracą systemu rozproszonego (sieci);
8. funkcji interpretera poleceń.

# Składowe moduły systemu

## **Moduł zarządzania procesami**

- tworzenie i usuwanie procesów (użytkowych oraz systemowych);
- wstrzymywanie i wznowianie procesów;
- udostępnianie mechanizmów synchronizacji procesów;
- udostępnianie mechanizmów komunikacji między procesorowej;
- udostępnianie mechanizmów obsługi blokad.

# Składowe moduły systemu

## **Moduł zarządzania pamięcią operacyjną**

- prowadzi ewidencję aktualnie wykorzystywanych części pamięci operacyjnej wraz z informacją o tym w czym są władaniu;
- rozstrzyga o tym w jaki sposób wykorzystać zwolnione obszary pamięci – jakiemu procesowi przydzielić zwolniony obszar pamięci;
- przydziela i zwalnia obszary pamięci stosowanie do potrzeb.

# Składowe moduły systemu

**Moduł  
zarządzania  
pamięcią  
pomocniczą  
(np.  
dyskami):**

- zarządza wolnymi obszarami pamięci;
- dokonuje przydziału pamięci;
- planuje przydział obszarów pamięci dyskowej.

# Składowe moduły systemu

## **Moduł zarządzania systemem wejścia- wyjścia**

- **Udostępnia procesowi użytkownika dostęp do urządzeń zewnętrznych za pośrednictwem systemu operacyjnego realizując bezkonfliktową komunikację z urządzeniami wejścia-wyjścia (wykorzystując do tego celu przerwania oraz system DMA).**

# Składowe moduły systemu

Plik jest logiczną jednostką służącą do przechowywania danych w pamięci zewnętrznej (np. na dysku).

## **Moduł zarządzania plikami:**

- tworzy oraz usuwa pliki oraz katalogi, w których przechowywane są pliki;
- udostępnia mechanizmy do manipulowania plikami i katalogami (np. usuwanie, czy kopiowanie);
- dokonuje odwzorowania plików na obszary pamięci pomocniczej;
- składa pliki na trwałych nośnikach.



# Usługi systemu operacyjnego

1. Realizacja procesu obliczeniowego (programu).
2. Obsługa operacji wejścia-wyjścia.
3. Manipulowanie systemem plików.
4. Komunikacja pomiędzy procesami realizowanymi współbieżnie.
5. Wykrywanie oraz obsługa błędów.
6. Gospodarka zasobami systemu.
7. Ochrona zasobów systemu komputerowego.
8. Ewidencja wykorzystania zasobów systemu komputerowego i rozliczanie użytkowników.

# Funkcje systemowe

- Funkcje (wywołania) systemowe stanowią interfejs pomiędzy realizowanym procesem (wykonywanym programem) a systemem operacyjnym.
- Funkcje systemowe są wywoływane w celu wykonania przez system operacyjny określonych zadań wynikających z realizacji procesu obliczeniowego.
- Funkcje systemowe są dołączane do kompilowanego programu w procesie kompilacji.

Dzielą się na funkcje:

- nadzorowania procesów,
- operacji na plikach,
- operacji na urządzeniach WE-WY,
- utrzymywania informacji oraz
- komunikacji.

# Rodzaje funkcji systemowych

1. Nadzorowanie procesów: zakończenie (*End*), zaniechanie (*Abort*), załadowanie (*Load*), wykonanie (*Execute*), oczekiwanie na zdarzenie (*Wait for Event*), itp.
2. Operacje na plikach: utworzenie pliku (*Create File*), usunięcie pliku (*Delete File*), otwarcie pliku (*Open*), zamknięcie (*Close*), czytanie (*Read*), pobranie atrybutów pliku (*Get File Attributes*), itp.
3. Operacje na urządzeniach: zamówienie urządzenia (*Request Device*), zwolnienie urządzenia (*Release Device*), czytanie (*Read*), pisanie (*Write*), pobranie atrybutów urządzenia (*Get Device Attributes*), itp.

# Rodzaje funkcji systemowych

4. Utrzymywanie informacji: pobranie daty lub czasu (*Get Time or Date*), ustawianie daty lub czasu (*Set Time or Date*), pobranie danych systemowych (*Get System Data*), pobranie atrybutów procesu (*Get Process Attributes*), itp.
5. Komunikacja: utworzenie, usunięcie połączenia komunikacyjnego (*Create, Delete Communication Connection*), nadawanie i odbieranie komunikatów (*Send, Receive Messages*), przekazanie informacji o stanie (*Transfer Status Information*), itp.

# Zarządzanie procesami

# Procesy

Procesem nazywa się każde działanie systemu komputerowego.

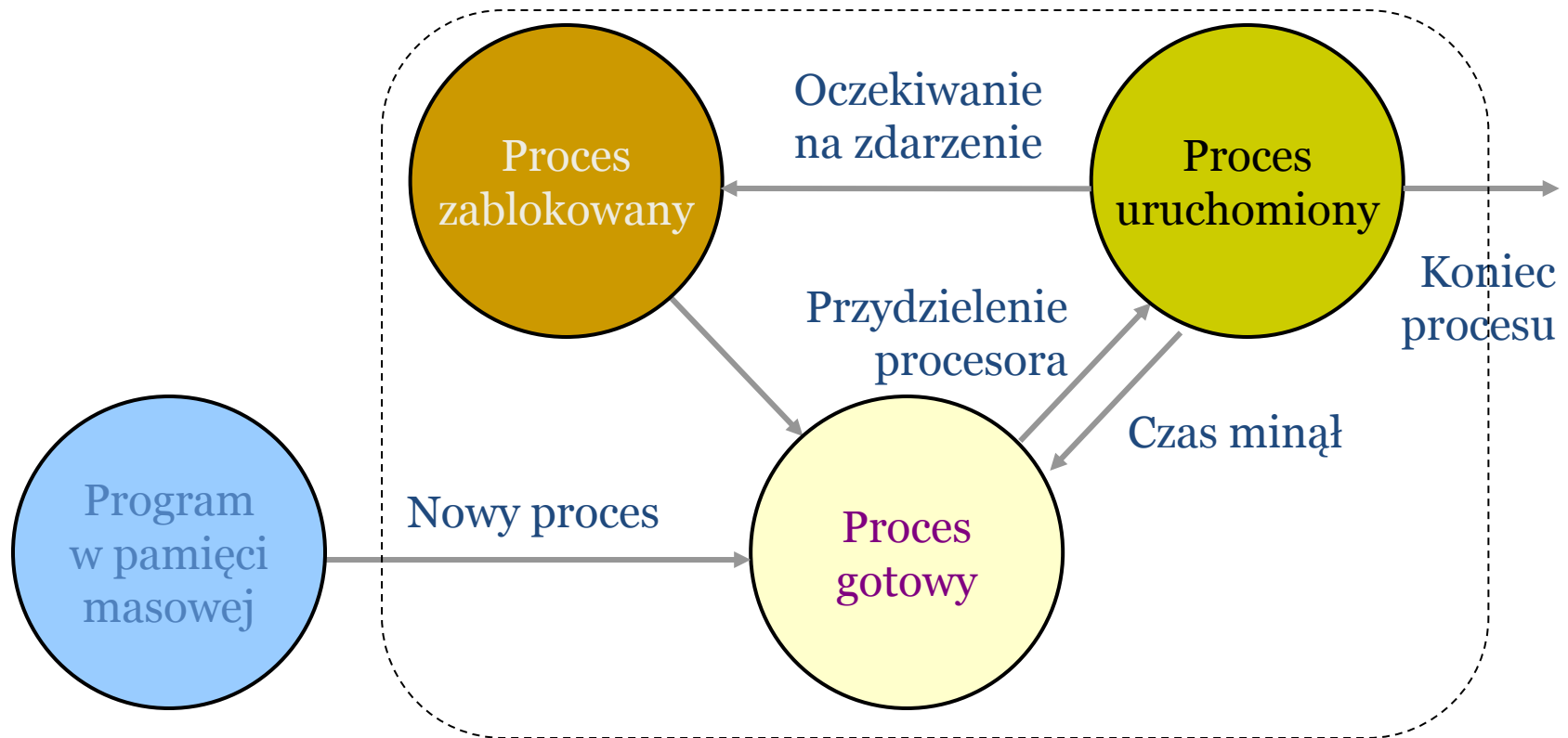
Procesy realizowane w komputerze wykonują programy, stąd potrzebują one niezbędnych zasobów, np. *czas procesora, pamięć operacyjna, pliki, urządzenia WE-WY.*

System komputerowy składa się ze zbioru procesów np. procesy S.O., procesy użytkownika.

# Proces sekwencyjny

- Proces sekwencyjny jest programem wykonywanym sekwencyjnie.
- W określonej chwili czasowej wykonywany jest jedynie jeden rozkaz kodu programu.
- Z procesem związane są również takie pojęcia jak:
  - stan procesu, czy
  - sekcja danych.
- Sam program nie jest procesem, gdyż jest obiektem pasywnym (podobnie jak plik w pamięci), zaś proces jest obiektem aktywnym.

# Stany procesu





# Blok kontrolny procesu

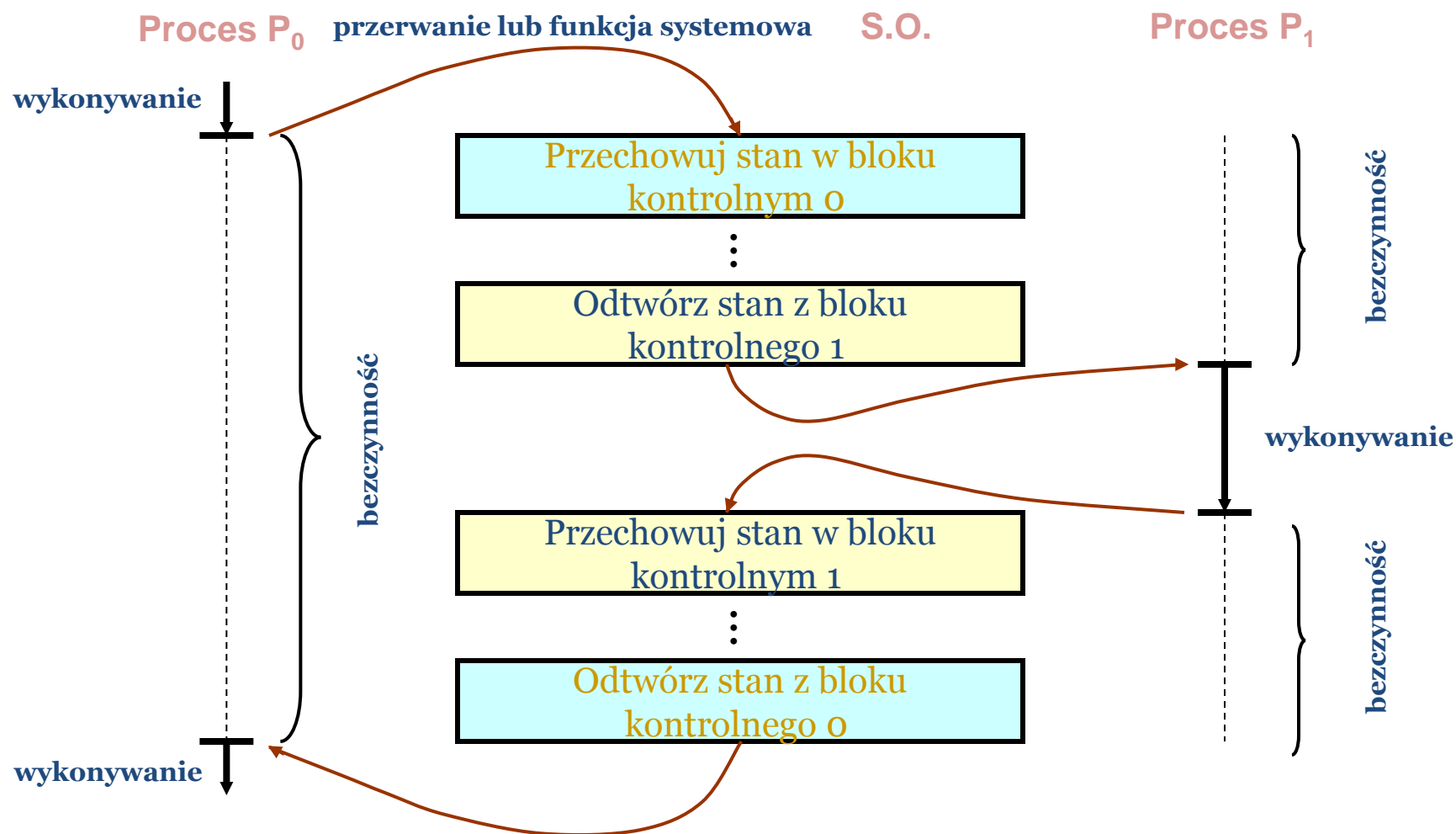
Każdy proces jest odwzorowany w S.O. poprzez przypisany jemu *blok kontrolny procesu* zawierający niezbędne dane dotyczące procesu:

- stan procesu;
- licznik rozkazów;
- rejestry procesora;
- informacje o planowaniu przydziału procesora;
- informacje o zarządzaniu pamięcią;
- informacje do rozliczeń;
- informacje o stanie urządzeń WE-WY.

# Blok kontrolny procesu

Wskaźnik	Stan procesu
Numer procesu	
Licznik rozkazów	
Rejestry	
Ograniczenia pamięci	
Wykaz otwartych plików	
.	
.	
.	

# Przełączanie procesora



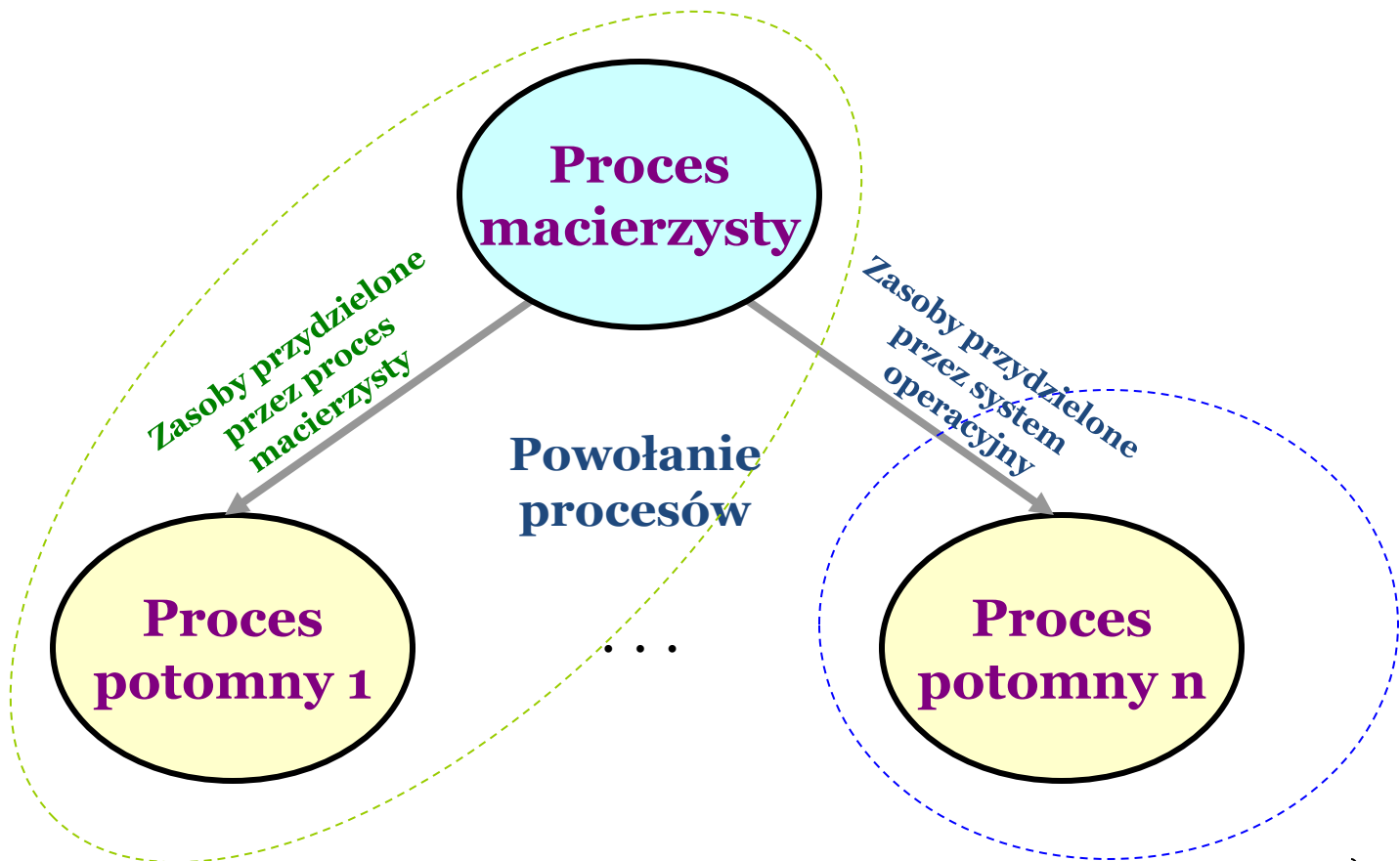
# Procesy współbieżne

Procesy współbieżne są to takie procesy, które współdzielą zasoby (moc obliczeniową) pomiędzy wiele procesów jednocześnie.

Korzyści ze współbieżności procesów:

- podział zasobów fizycznych (procesor, WE-WY);
- podział zasobów logicznych (pliki);
- multiplikacja (przyspieszenie) obliczeń (w przypadku systemów wieloprocessorowych);
- modularność (łatwa dekompozycja procesów);
- wygoda w użytkowaniu.

# Tworzenie procesu



1. Proces macierzysty jest wykonywany współbieżnie z potomkami.
2. Proces macierzysty jest wstrzymany dopóki procesy potomne nie zakończą pracy.

Zasady  
pracy

# Zakończenie procesu

Przyczyny zakończenia procesów:

- Proces zakończył działanie i poinformował S.O. o stanie zakończenia działań.
- Proces potomny przekroczył zakres przydzielonych mu zasobów.
- Dalsza realizacja procesu obliczeniowego potomka stała się zbędna.
- Kończący się proces macierzysty wymusza zakończenie procesów potomnych (zakończenie kaskadowe).

# Zależności pomiędzy procesami

<i>Proces NIEZALEŻNY</i>	<i>Proces WSPÓŁPRACUJĄCY</i>
<ol style="list-style-type: none"><li>1. Na stan procesu nie wpływa żaden inny proces.</li><li>2. Działanie procesu jest deterministyczne, czyli wynik zależy jedynie od stanu wejścia.</li><li>3. Wynik jest powtarzalny, czyli działanie da się powielać.</li><li>4. Działanie może być wstrzymywane i wznowiane dowolną ilość razy bez żadnych skutków</li></ol>	<ol style="list-style-type: none"><li>1. Oddziałuje na inne procesy.</li><li>2. Stan procesu jest dzielony z innymi procesami.</li><li>3. Wynik działania procesu jest zależny od stanu innych procesów.</li><li>4. Wynik działania jest niedeterministyczny i może być różny dla tych samych stanów wejściowych.</li></ol>