Organizacja i architektura komputerów [1]

Wykład 10

Piotr Patronik

22 maja 2015

[1] (Prawie) dokładna kopia slajdów dr hab inż. J. Biernata

Zarządzanie pamięcią (memory management)

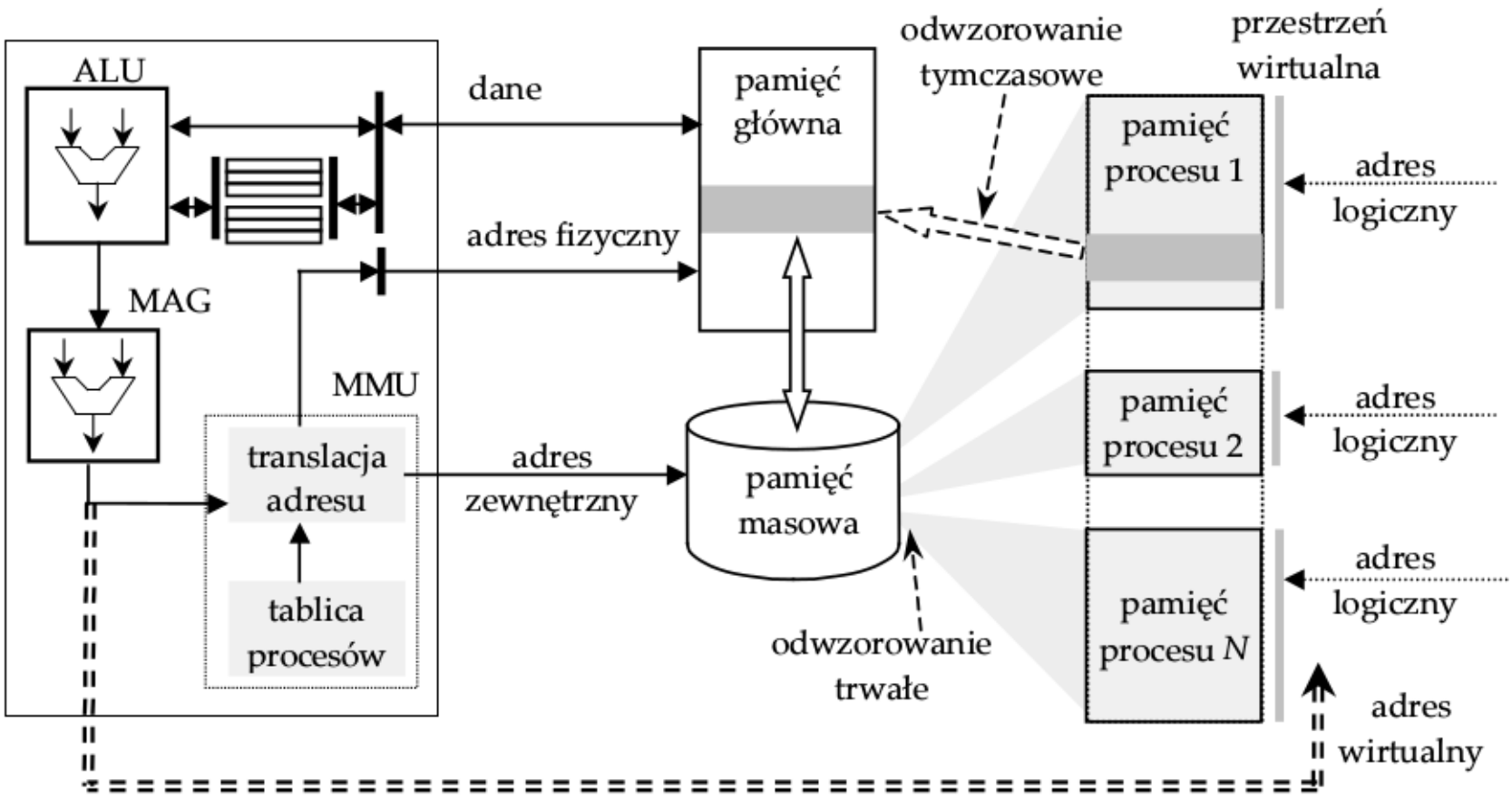
Funkcje zarządzania pamięcią

- przydział zasobów pamięci (*memory allocation*)
- ochrona zasobów pamięci (*memory protection*)
- współdzielenie obszarów pamięci (*memory sharing*) przez różne procesy
- przemieszczanie obszarów pamięci (*memory relocation*)
- przeźroczysta organizacja pamięci fizycznej i logicznej
  - relacje logiczne danych w programie niewrażliwe na zarządzanie
  - elastyczne powiązanie logicznych struktur danych z fizycznymi lokacjami, realizowane dynamicznie

Pamięć wirtualna

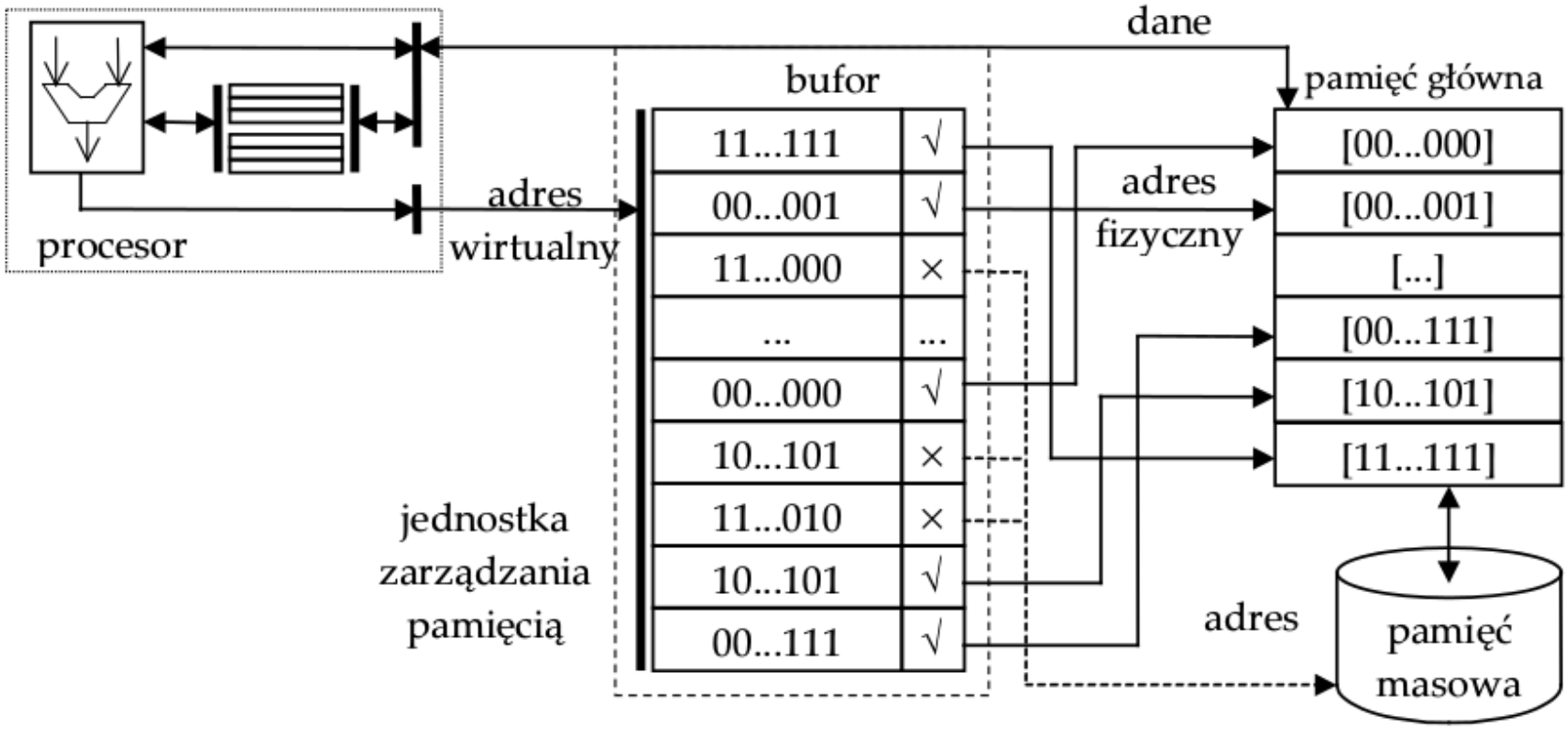
Wirtualna przestrzeń adresowa – suma logicznych przestrzeni pamięci (widzianych w programie stanowiącym treść procesu)
translacja adresu wirtualnego (logicznego) na adres rzeczywisty
odwzorowanie adresu wirtualnego w pamięci fizycznej (*real memory*).

Koncepcja pamięci wirtualnej



Adres logiczny, wirtualny i fizyczny

Translacja adresu



Translacja adresu w układzie zarządzania pamięcią

Zasada lokalności

W określonym przedziale czasu program przejawia tendencję do grupowania odwołań do małego fragmentu dostępnej przestrzeni adresowej

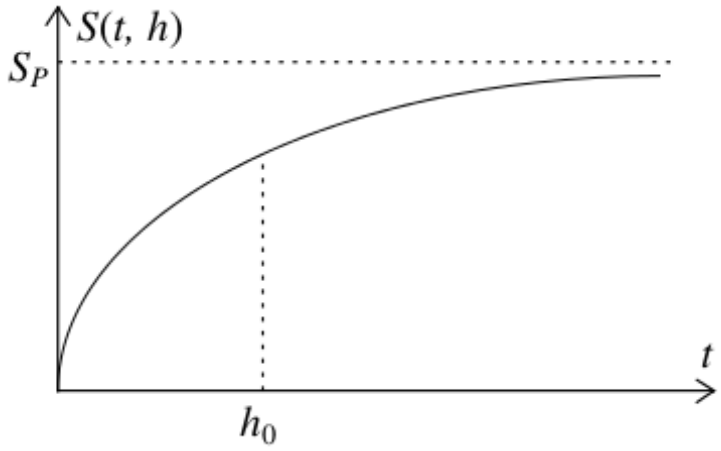
Lokalność czasowa – tendencja do powtarzania odwołań, realizowanych w niedawnej przeszłości (realizacja pętli, referencje do tablicy).

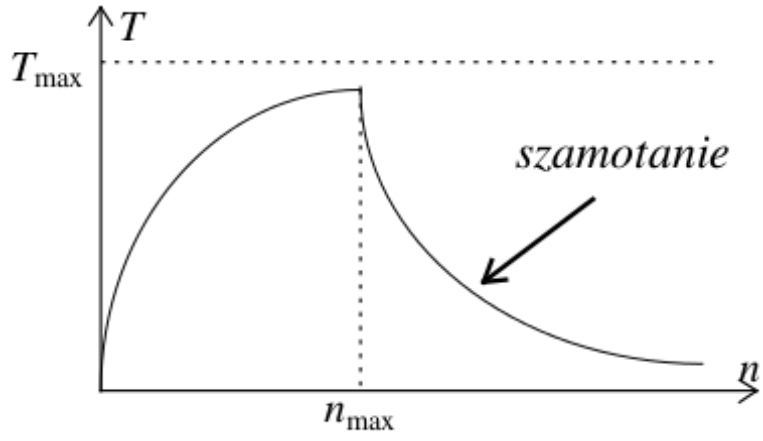
Lokalność przestrzenna – tendencja do odwołań do obiektów w obszarze adresowym obejmującym obiekty wcześniej użyte w programie (kolejne rozkazy programu, elementy regularnej struktury danych).

```
        mov eax, tablica[4*ebx]   ; szukanie najmniejszego w tablicy
label:  dec ebx                   ; liczb naturalnych
        cmp eax, tablica[4*ebx]
        jbe hop                   ; omiń gdy (eax) ⩽ (tablica[4*ebx])
        mov eax, tablica[4*ebx]
        mov adres, ebx
hop:    cmp ebx, 0
        jnz label
```

Model zbioru roboczego

Zbiór roboczy – zapotrzebowanie procesu na pamięć w okresie wykonania





Rozmiar zbioru roboczego $S(t, h)$ i przepustowość przetwarzania

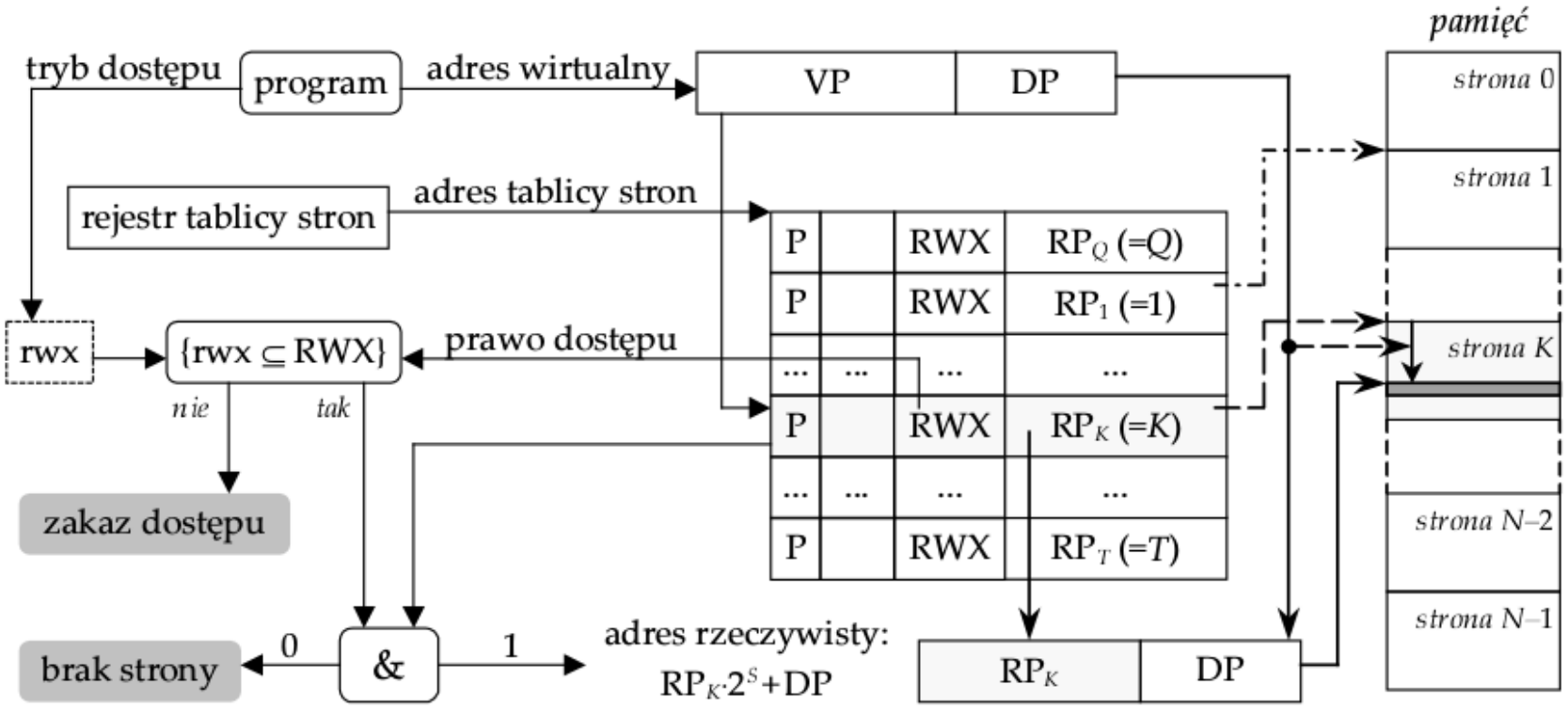
Efekt szamotania

Suma zbiorów roboczych procesów aktywnych $>$ rozmiar dostępnej pamięci

Heurystyka

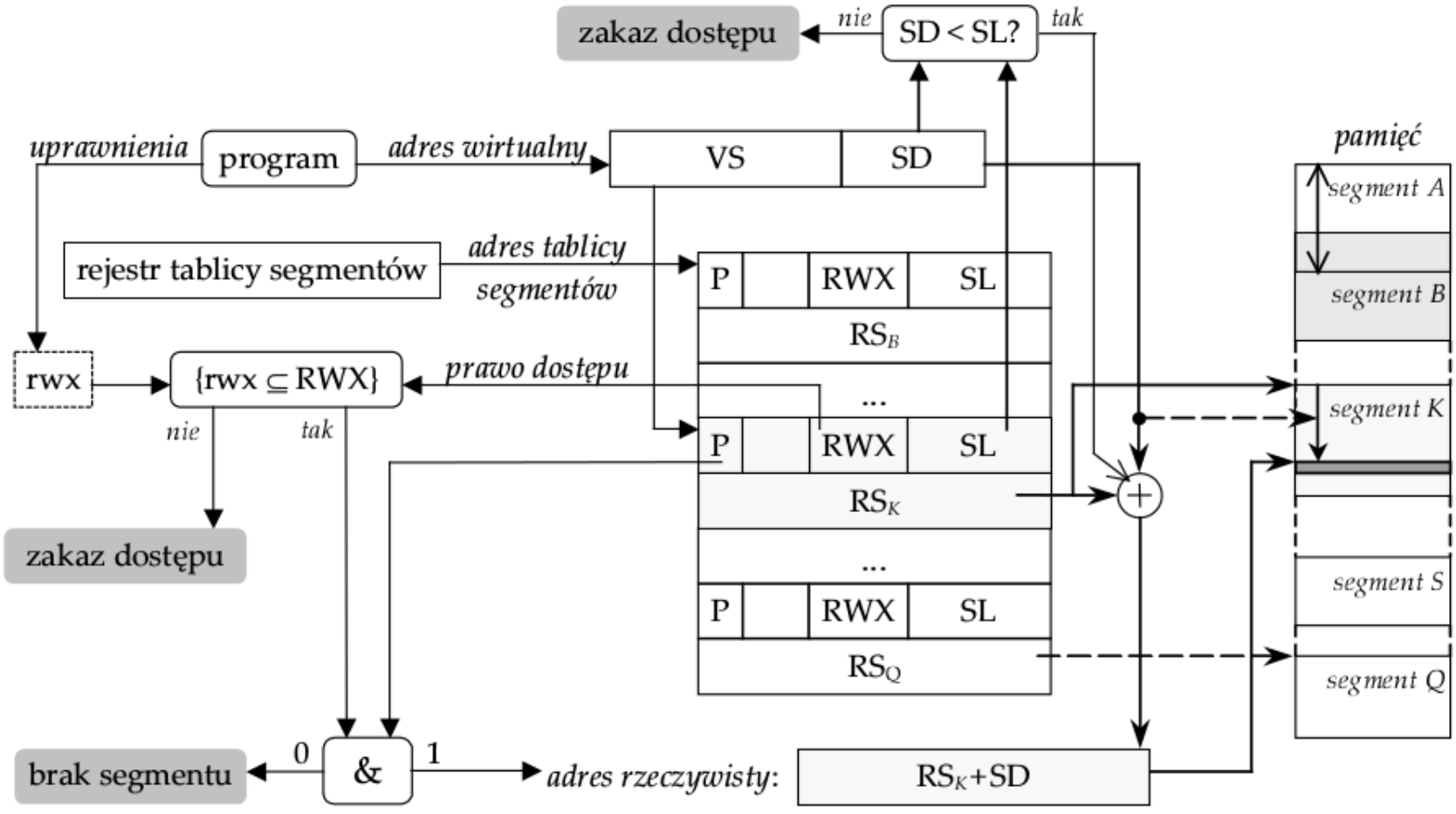
Nie wymieniaj bloku, który jest częścią zbioru roboczego aktywnego procesu i nie uaktywniaj procesu, którego zbiór roboczy nie może zostać w całości odwzorowany w pamięci głównej.

Stronicowanie



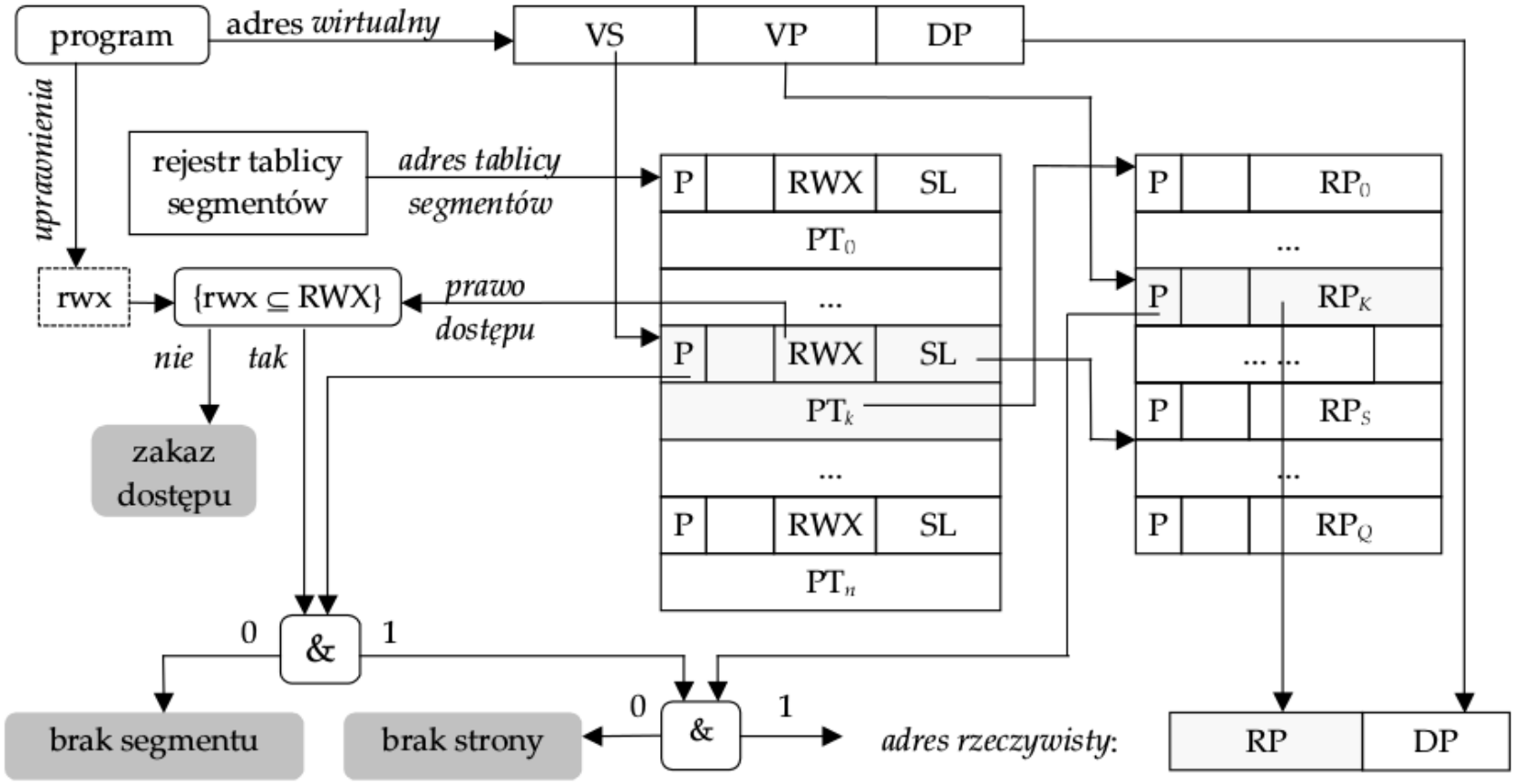
P – bit obecności strony, RWX – kod praw dostępu, VP – numer strony wirtualnej, RP – numer strony rzeczywistej, DP – przemieszczenie na stronie

Segmentacja



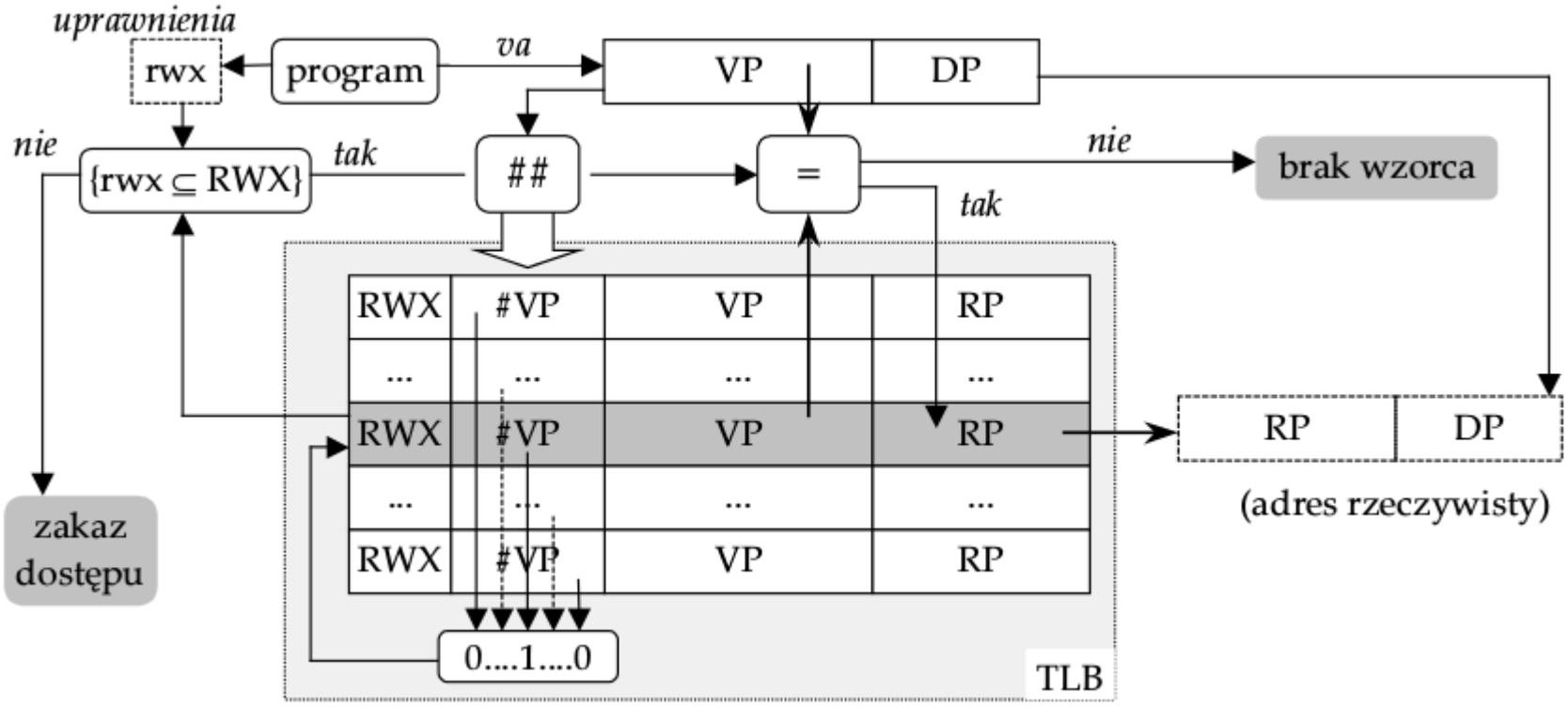
P – bit obecności, RWX – prawo dostępu, VS – numer segmentu wirtualnego, RS – adres rzeczywisty, SD – przemieszczenie w segmencie, SL – rozmiar

Segmentacja stronicowana



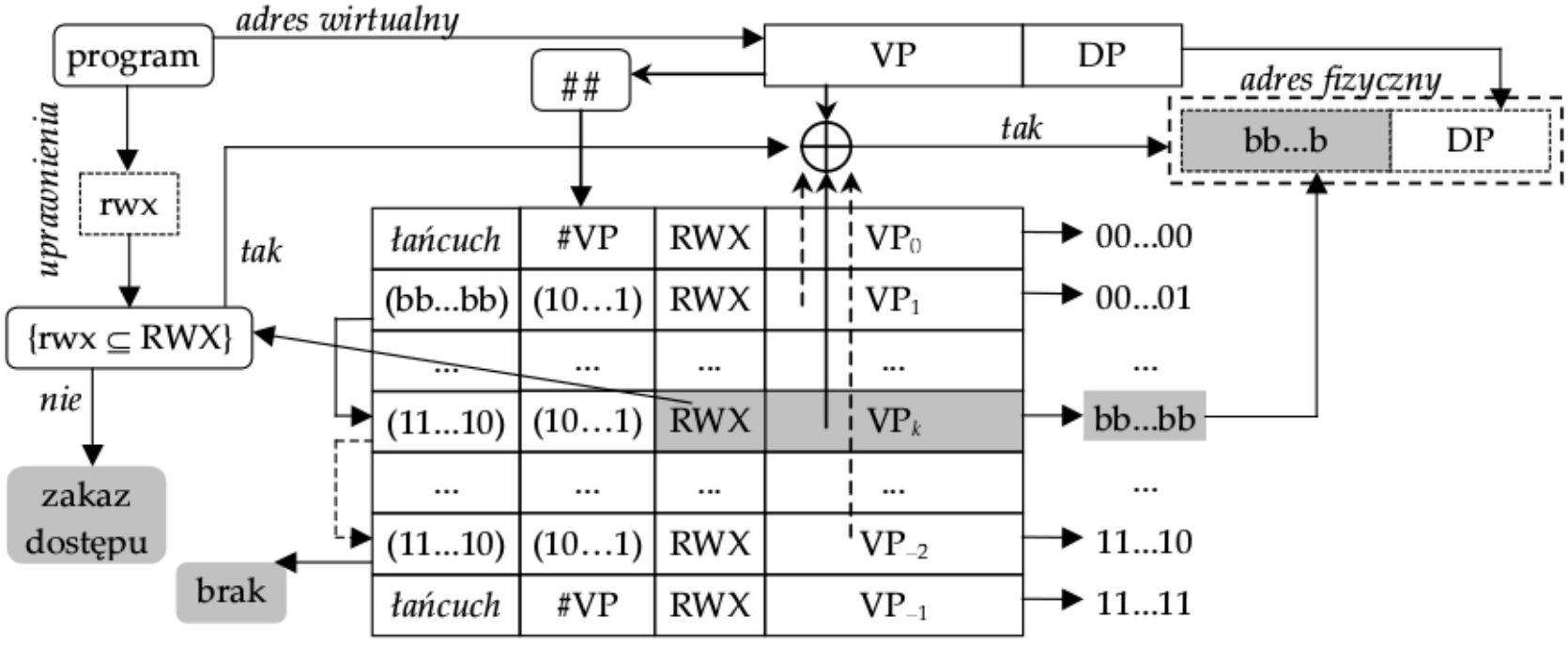
Translacja adresu w trybie segmentacji stronicowanej

Mechanizmy przyśpieszania translacji adresu



Bufor antycypacji translacji TLB (translation lookaside buffer) (va – adres wirtualny, VP, #VP – wirtualny numer strony i jego skrót, RP – rzeczywisty numer strony, DP – adres na stronie, RWX – kod praw dostępu)

Odwrócona tablica stron



VP – numer strony wirtualnej #VP – skrót, DP – adres na stronie skrót (hash value) – odwzorowanie wirtualnego adresu strony za pomocą funkcji mieszającej (hashing function) czas przeszukiwania – bardzo mało zależny od liczby stron rzeczywistych

Strategie zarządzania pamięcią

- strategie pobierania (*fetch policy*) – decyzje, kiedy załadować informację do pamięci głównej
- strategie przydziału (*placement policy*) – reguły i algorytmy wpasowania bloków informacji w wolne obszary pamięci głównej
- strategie wymiany (*relocation policy*) – reguły i algorytmy usuwania informacji z pamięci głównej.

Strategie pobierania

- pobranie wymuszone (*demand fetching*) na skutek błędu braku obiektu (*missing-item fault*)
- pobranie antycypowane (*prefetching*) na podstawie prognozy zapotrzebowania procesu na dane (zasady lokalności).

Strategie przydziału

- w pamięci stronicowanej – trywialne, rozmiar strony ustalony problem – wewnętrzna fragmentacja pamięci
- w pamięci segmentowanej – wpasowanie segmentów o różnych rozmiarach problem – zewnętrzna fragmentacja pamięci (dziury)

Przydział pamięci segmentowanej (1)

segment umieszczany w pierwszej dziurze o wystarczającym rozmiarze

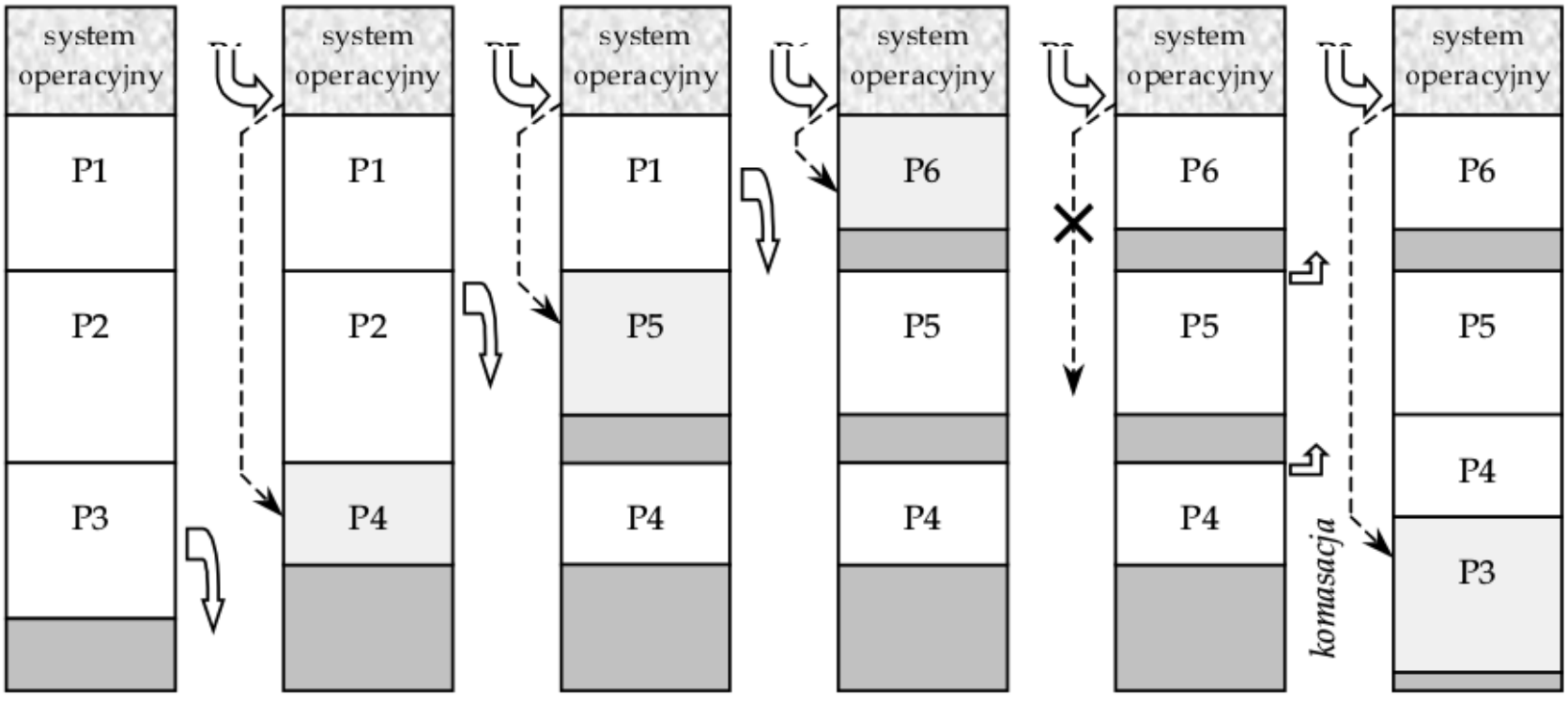
Metody:

- BF – *najlepsze wpasowanie* (*best fit*) – lista uporządkowana według rosnących adresów
- WF – *najgorsze wpasowanie* (*worst fit*) – lista uporządkowana według malejących adresów
- FF – *pierwsze wpasowanie* (*first fit*) – lista uporządkowana według rosnących adresów, przeszukiwanie może być cykliczne (wznawialne)
- BB – *wpasowanie binarne* (*binary buddy*) – są tworzone listy dziur o rozmiarach $[2^{ip}, 2^{(i+1)p}]$, wpasowanie segmentu metodą FF w obrębie listy, kolejność adresów na listach jest liniowa.

Problemy

- aktualizacja listy dziur
- *defragmentacja* pamięci (komasacja dziur)

Zewnętrzna fragmentacja partycji spójnych



Fragmentacja partycji (i pamięci segmentowanej)

Partycje (1)

Procesowi jest przydzielana część adresowalnego obszaru pamięci głównej.

$\longrightarrow$ intensyfikacja błędu braku bloku

Strategie przydziału obszaru pamięci procesom (*memory allocation*)

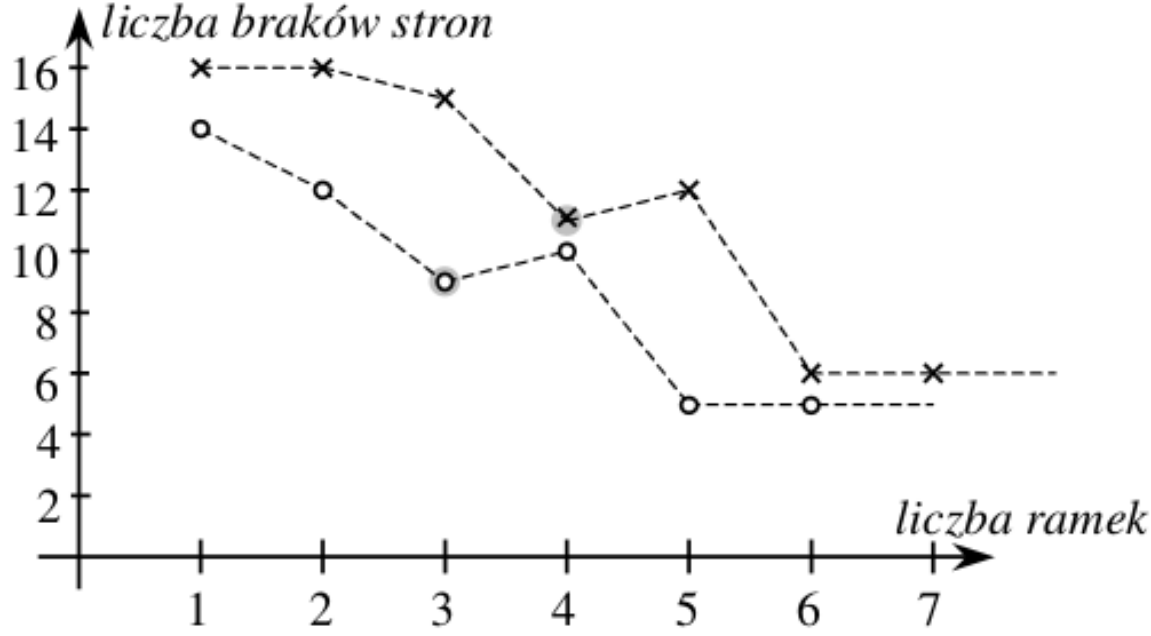
- partycja stała (*fixed–size partition*) – rozmiar obszaru pamięci przydzielonej procesowi jest stały w czasie życia procesu
- partycja zmienna (*variable–size partition*) – dynamiczny przydział pamięci, odpowiednio do aktualnych potrzeb procesu.

Strategie wymian dla stałych partycji:

- losowa (*random replacement*) – wyłącznie w środowisku programowym, w którym lokalność jest niewielka (np. bazy danych)
- FIFO (*first-in, first-out*) – kolejka bloków do wymiany jest ustawiana zgodnie z kolejnością ich umieszczania w pamięci; uwzględniana jest lokalność bloków, nie uwzględnia się intensywności ich używania
- FINUFO (*first-in, not used, first-out*) – każde wejście do kolejki ma znacznik używalności, kolejka przesuwa się cyklicznie
- LRU (*least recently used*) – wymienia się blok najdawniej używany.

Partycje (2)

Anomalia Belady'ego – częstość błędu braku strony nie jest monotoniczną funkcją rozmiaru przydzielonego obszaru (liczby stron) i osiąga lokalne minimum dla pewnej niewielkiej liczby stron (FIFO)



sekwencje odwołań 1,2,3,4,5,1,2,3,6,1,2,3,4,5,6,4 oraz 1,2,3,4,1,2,5,1,2,3,4,5,4

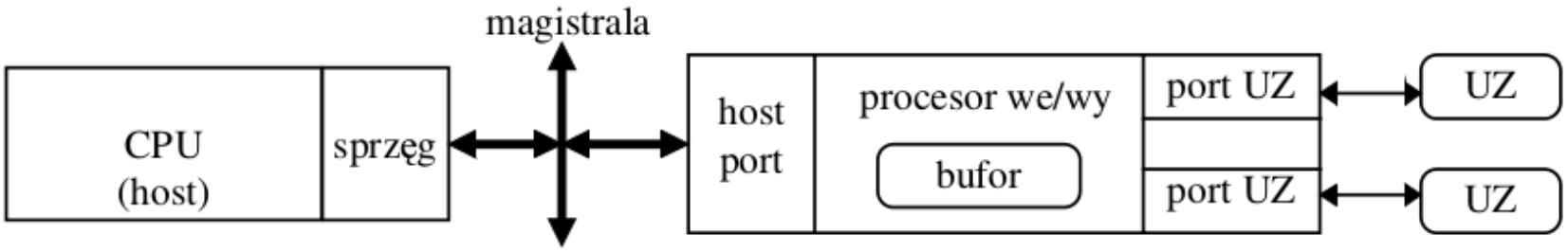
Partycje (3)

Strategia optymalna (MIN) – wymiana bloku, który będzie użyty najpóźniej
teoretyczna, wymaga antycypacji kolejności wymian
Strategie wymian dla partycji zmiennych (przy stronicowaniu):

- WS – wymiana całego zbioru roboczego (*working set replacement*)
- PFF – wymiana stosownie do częstości występowania błędu braku strony (*page fault frequency*) – ustala się wartość progową PFF i jeżeli częstość błędu braku strony $pff < \text{PFF}$, to wymieniane są wszystkie strony nieużywane od ostatniej wymiany, jeśli zaś $pff > \text{PFF}$, to nie jest dokonywana wymiana, lecz zwiększany jest rozmiar partycji.

Strategia optymalna VMIN – wymiana bloku, który będzie użyty najpóźniej, lecz z możliwością zmiany rozmiaru partycji.
teoretyczna, wymaga antycypacji kolejności wymian

Obsługa wejścia i wyjścia



Podsystem wejścia/wyjścia

Urządzenia we/wy:

- magazynujące (*storage*) – przechowywanie danych
  - pamięci wtórne i tercjalne, archiwizery
- gromadząco-rozsyłające (*source/sink*) – konwersja i rozsyłanie danych
  - komunikacja człowieka z komputerem: mysz, klawiatura, monitor ekranowy, drukarka
  - sprzęg procesów przemysłowych z komputerem: czujnik, regulator, konwerter A/C i C/A, sterownik
  - komunikacja między inteligentnymi terminalami: modem, łącze sieci komputerowej, łącze transmisji szeregowej lub równoległej

Cechy eksploatacyjne urządzeń zewnętrznych

- czas dostępu (*access time*), zwłoka dostępu (*latency*) – znacznie ($> 10^5$ razy) większe niż dla pamięci (dostęp sekwencyjny)
- przepustowość (*bandwith*) – maksymalna liczba danych przesyłanych w kwancie czasu (transmisja równoległa z szybkością GB/s)
- ryzyko błędu (*error rate*) – średnia częstość występowania błędów ($< 10^{-6}$)

Sterowniki

- łącza bezpośredniego (USB) – sprzęg (*interface*) we/wy
  - szeregowego (protokół RS232, RS485, USB)
  - równoległego (protokół Centronics)
- magistral dedykowanych
- magistrali współdzielonej

Sterowanie urządzeniami

- programowanie sterownika (wysyłanie poleceń)
- testowanie sterownika
- obsługa przerwań sygnalizowanych przez urządzenia
- obsługa błędów urządzeń

Oprogramowanie we/wy

Struktura warstwowa (*software layers*)

- „uchwyty przerwań" (*interrupt handlers*) – procedury obsługi zgłoszeń
- sterowniki urządzeń (*device drivers*) – działania specyficzne dla urządzeń (translacja poleceń logicznych („odczytaj blok danych") na zestaw poleceń („odczytaj sektor N na ścieżce M w dysku H")):
  - buforowanie (kolejkowanie) poleceń logicznych
  - zmiana porządku poleceń w celu poprawy przepustowości transmisji
  - zmiana logicznych adresów urządzeń i bloków danych na adresy fizyczne
  - obsługa błędów transmisji
- funkcje użytkownika – oprogramowanie niezależne od urządzeń:
  - zmiana symbolicznej nazwy urządzenia (standardowe wejście i wyjście, drukarka) na nazwę logiczną (adres bloku sterującego procesu obsługi)
  - przeformatowanie danych na postać wymaganą przez urządzenie (buforowanie, upakowanie, rozpakowanie, translacja kodu danych)
  - przydział i zwolnienie przydziału pamięci
- uaktywnienie (proces użytkownika) – określenie parametrów transmisji, przygotowanie danych i zainicjowanie komunikacji we/wy

Procesy wejścia i wyjścia

Sterowniki urządzeń (*device drivers*) – zasoby chronione
Funkcje obsługi wejścia / wyjścia – osobne procesy na poziomie nadzoru
Proces obsługi wejścia lub wyjścia:

- program wykonywany przez CPU
- program wykonywany przez sterownik

Klasyfikacja

- bezpośrednie we/wy (*direct I/O*) – funkcje sterownika wykonuje procesor: testowanie statusu urządzenia (*busy waiting*) i nadzór wykonania poleceń
- nakładane (*overlapped*) we/wy – obsługa w trybie przerwań precyzyjnych, wymaga intensywnej synchronizacji
- autonomiczne (*autonomous*) we/wy – obsługa w trybie DMA (bezpośredni dostęp do pamięci), wymaga minimum synchronizacji

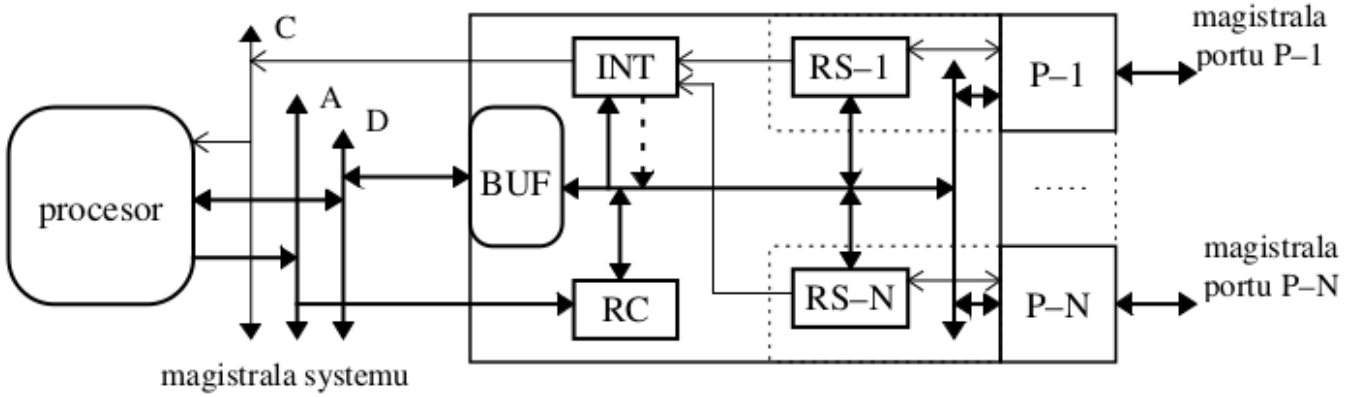
Nakładane we/wy (overlapped I/O)

Synchronizacja procesu obsługi we/wy w trybie przerwań sygnalizujących

- gotowość urządzenia we/wy do transmisji
- zakończenie operacji
- wystąpienie wyjątku sygnalizowanego (na przykład błąd transmisji)

Procesy we/wy są niezależne

- wiele urządzeń obsługiwanych jednocześnie, problem identyfikacji źródła
- konieczność przełączania procesów

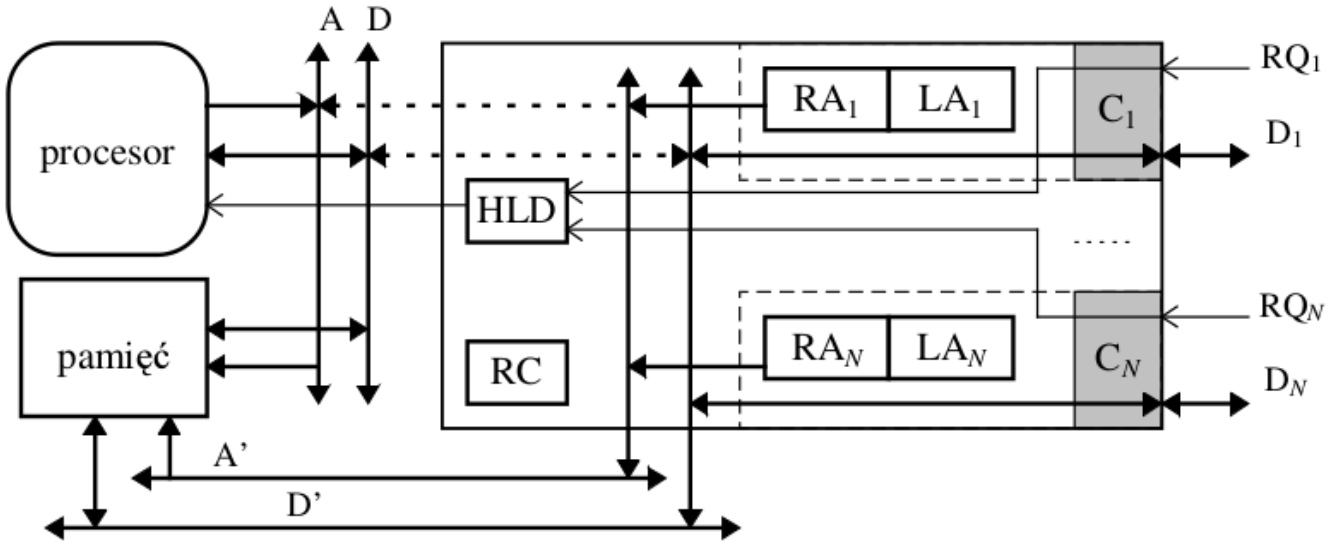


Realizacja nakładanego we/wy

Autonomiczne we/wy

Transmisja bloku danych zamiast transmisji danych pojedynczych

- zmniejszenie częstości synchronizowania procesów
- redukcja narzutów czasowych synchronizowania
- konieczne bufory danych – najprościej w pamięci głównej
  $\longrightarrow$ transmisja z pominięciem CPU (DMA)



Realizacja autonomicznego we/wy

Bezpośredni dostęp do pamięci (DMA)

Obsługa w trybie bezpośredniego dostępu do pamięci

- programowanie procesora DMA – podanie parametrów transmisji: adresu źródłowego i docelowego danych, rozmiaru bloku i protokołu transmisji
- zainicjowanie transmisji – skutek zgłoszenia żądania transmisji przez kanał DMA (procesor DMA ma zwykle kilka niezależnych kanałów)

Procedura

- zgłoszenie w kanale DMA żądania transmisji (*bus request*, *hold*)
- potwierdzenie udostępnienia magistral (*bus grant*, *hold acknowledge*)
- wykonanie transferu DMA i zwolnienie magistral

Transfer pojedynczy (*single-cycle DMA*) – wykradanie cykli (cycle–stealing)

Transfer blokowy (*burst-mode DMA*)

- blokowanie dostępu do magistral – pamięci dwuportowe
- różny rozmiar bloków danych w urządzeniach uczestniczących