Zarządzanie pamięcią (*memory management*)

Cel: wspomaganie **bezpiecznego wykonania współbieżnych procesów**

Funkcje zarządzania pamięcią

- przydział zasobów pamięci (ang. *memory allocation*)
- ochrona zasobów pamięci (ang. *memory protection*)
- współdzielenie obszarów pamięci (ang. *memory sharing*) przez różne procesy
- przemieszczanie obszarów pamięci (ang. *memory relocation*)

Postulat: przeźroczysta organizacja pamięci fizycznej i logicznej

- relacje logiczne danych w programie niewrażliwe na zarządzanie
- elastyczne powiązanie *logicznych struktur* danych z *fizycznymi lokacjami*, realizowane dynamicznie (w trakcie wykonania procesu)

Koncepcja pamięci wirtualnej

Problem:

Elastyczne (niezależne od sprzętu) powiązanie
logicznej przestrzeni programu i **fizycznej przestrzeni pamięci wtórnej**
z **pamięcią operacyjną** komputera

Rozwiązanie:

Powiązanie pośrednie:
przestrzeń operacyjna $\Leftrightarrow$ jakaś przestrzeń adresowa
+ jakaś przestrzeń adresowa $\Leftrightarrow$ przestrzeń wtórna (dysk)

Przestrzeń logiczna jest pierwotna (zdefiniowana przez programistę!!)
Istnieje powiązanie (kompilacja+konsolidacja):
przestrzeń operacyjna (fizyczna) $\Leftrightarrow$ logiczna przestrzeń adresowa (program)

Wniosek:
jakaś przestrzeń adresowa $\leftrightarrow$ zebrane logiczne przestrzenie adresowe procesów

Koncepcja pamięci wirtualnej

Wirtualna przestrzeń adresowa

– złożenie logicznych przestrzeni adresowych opisanych w programie (który stanowi treść procesu)

Architektura klasyczna (von Neumanna):

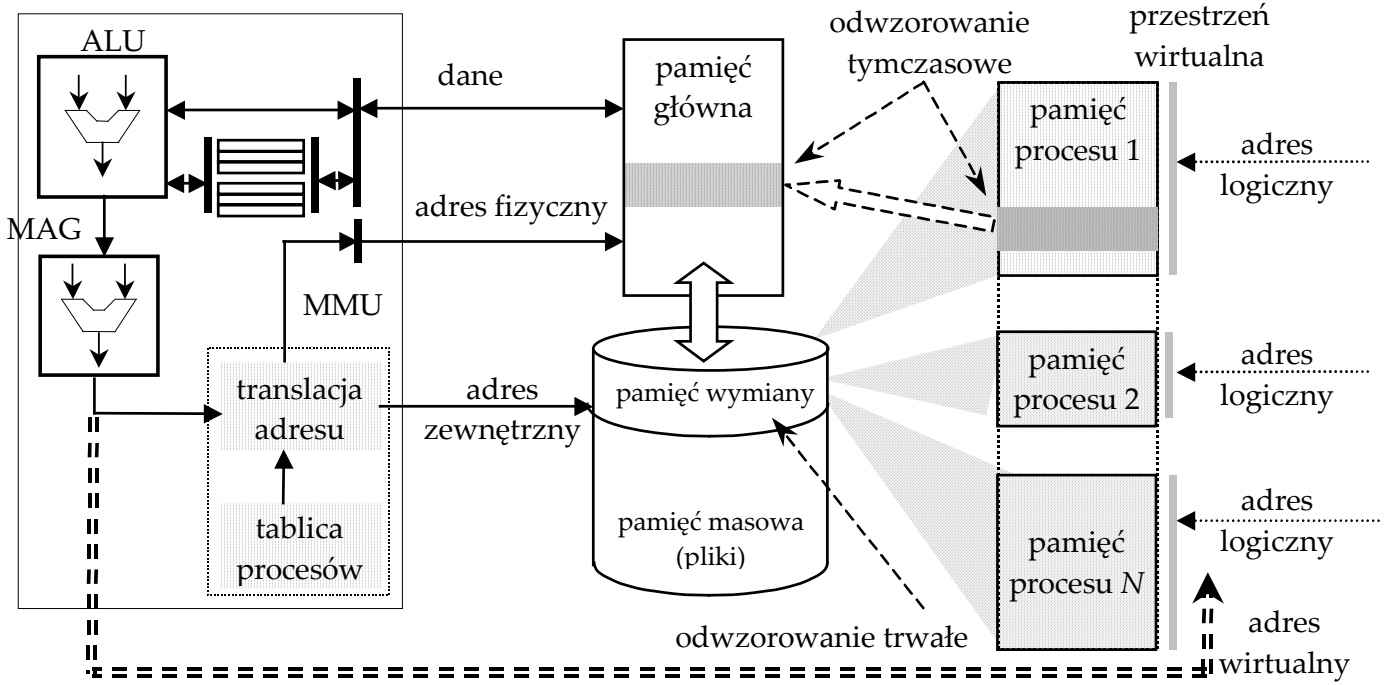
- z poziomu instrukcji dostęp tylko do zmiennych w pamięci operacyjnej
- *odwzorowanie* zmiennych procesu w pamięci operacyjnej (*real memory*)
- obserwowalna lokalność danych w skali makro – segmenty programu
    - lokalność → odwzorowanie bloków (stron lub segmentów)

- dane (w tym kody) wielu programów można przechować w pamięci wtórnej
- z poziomu instrukcji nie ma dostępu do pamięci wtórnej

konieczne odwzorowanie:

- kopiowanie danych z pamięci wtórnej do pamięci operacyjnej
- *translacja* adresu *wirtualnego* (*logicznego*) na adres *rzeczywisty*
    - funkcja dyskretna RA=*f*(VA) – realizacja: tablica par {VA,RA}

Koncepcja pamięci wirtualnej a hierarchia pamięci



Adres logiczny (struktury danych programu), wirtualny (VA) i fizyczny (RA)
odwzorowanie przestrzeni wirtualnej **w** pamięci głównej: RA = *f* (VA)

Odwzorowania przestrzeni wirtualnej

Odwzorowanie *spójnego* bloku pamięci opisuje *funkcja dyskretna* typu „**w**":

f: adres wirtualny | rozmiar (VA | S) $\rightarrow^{w}$ adres fizyczny (RA)

realizacja: tablica par (VA | S, RA)

Zasada lokalności – odwzorowanie bloków danych o ciągłym adresowaniu

- jedna reguła dostępu dla wszystkich obiektów wewnątrz bloku
- rozmiar bloku – niezmienny podczas ważności odwzorowania
- adres względny – niezmienny podczas ważności odwzorowania

odwzorowanie spójnych struktur logicznych – segmentacja

- różne rozmiary bloków – konieczna weryfikacja poprawności adresu
- adres początku bloku wirtualnego → adres początku bloku fizycznego

odwzorowanie bloku fizycznego – stronicowanie

- (rozmiar bloku ustalony – rozmiar 2^k upraszcza odwzorowanie)
- numer bloku jednoznacznie wyznacza adres początku bloku
- numer bloku wirtualnego → numer bloku fizycznego
  (numer strony → numer ramki stron)

Schematy zarządzania pamięcią

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

Schematy pobierania

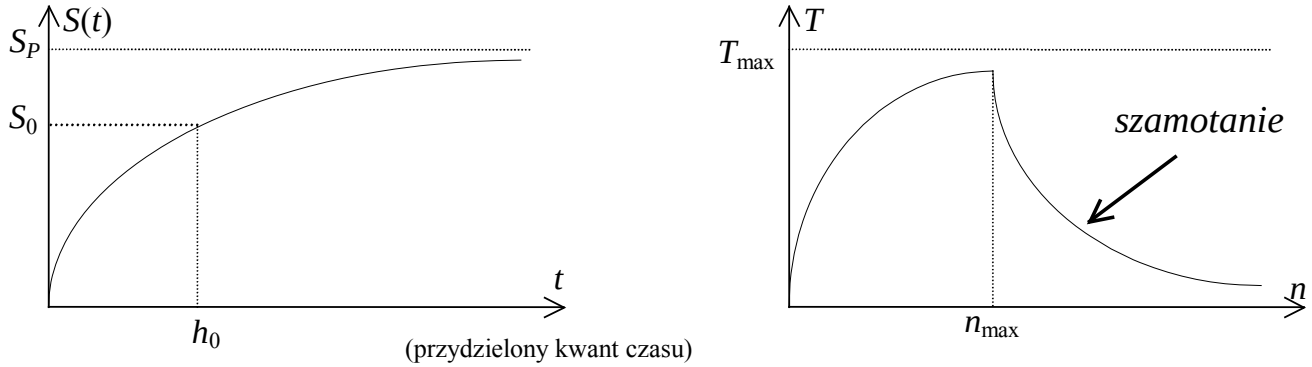
- pobranie wymuszone (ang. *demand fetching*) na skutek *błędu braku obiektu* (ang. *missing–item fault*)
- pobranie antycypowane (ang. *prefetching*) na podstawie prognozy zapotrzebowania procesu na dane (zasady lokalności).

Schematy 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)

Model zbioru roboczego

Zbiór roboczy – zapotrzebowanie procesu na pamięć w stanie wykonania



Rozmiar zbioru roboczego $S(t)$ 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 operacyjnej (głównej).

Odwzorowanie bloków przestrzeni wirtualnej w pamięci operacyjnej

Odwzorowanie pamięci procesu (przestrzeni wirtualnej) w pamięci operacyjnej:

- *totalne* – zajmuje cały dostępny obszar pamięci (oprócz zastrzeżonych: jądro s.o., procedury rezydentne, systemowe struktury danych)
- *częściowe* – zajmuje część dostępnego obszaru pamięci

Odwzorowanie częściowe wystarcza, co wynika z *lokalności odwołań*:

czasowej – tendencja do powtarzania odwołań,

przestrzennej – tendencja do grupowania odwołań w obszarze adresowym obejmującym obiekty wcześniej adresowane

Partycjonowanie

Jeśli można oszacować przeciętne zapotrzebowanie procesu na pamięć fizyczną (rozmiar zbioru roboczego), to jest możliwe

bezkonfliktowe jednoczesne odwzorowanie pamięci kilku procesów.

Partycja – część pamięci głównej przydzielonej procesowi.

Partycje - przydział pamięci

Zapotrzebowanie – wynik akcji procesora:

- żądanie pobrania kodu rozkazu (skok/rozgałęzienie)
- żądanie dostarczenia danych aktualnie niedostępnych (odczyt pamięci)

Przydział statyczny – partycje stałe

- jednakowy rozmiar każdej partycji = średnie zapotrzebowanie procesów
- rozmiar dostosowany do wymagań procesu – zbiór roboczy

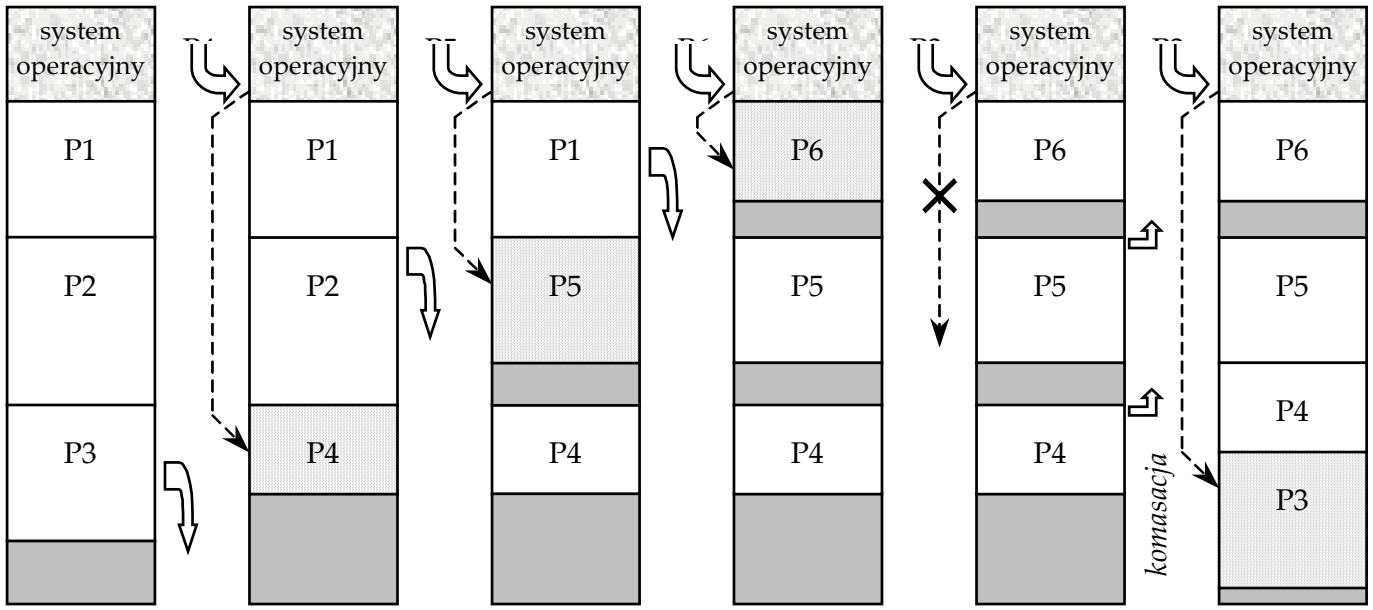
Przydział dynamiczny – partycje zmienne

- wstępny przydział – zbiór roboczy
- korekta zgodnie z faktycznym zapotrzebowaniem

Rozmiar partycji

- całkowita liczba bloków ustalonej wielkości (ramki stron)
- dowolny, dostosowany do potrzeb

Zewnętrzna fragmentacja partycji spójnych



Fragmentacja partycji (i pamięci segmentowanej)

Stronicowanie pamięci umożliwia tworzenie *partycji rozproszonych*

Partycje (1)

Procesowi jest przydzielana część adresowalnego obszaru pamięci głównej.

$\rightarrow$ intensyfikacja błędu braku bloku

Strategie *przydziału obszaru pamięci* procesom (ang. *memory allocation*)

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

Strategie wymian dla stałych partycji:

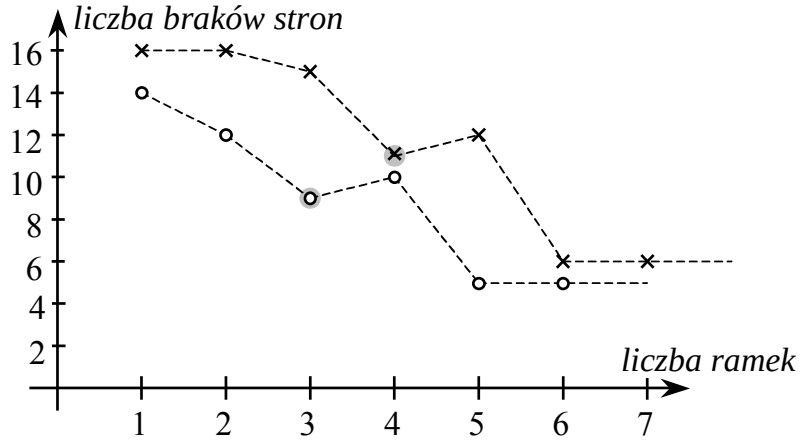
- losowa (ang. *random replacement*) – wyłącznie w środowisku programowym, w którym lokalność jest niewielka (np. bazy danych)
- FIFO (ang. *first–in, first–out*) – kolejka bloków do wymiany jest ustawiana zgodnie z kolejnością ich umieszczania w pamięci; uwzględniana jest lokalność bloków, nie uwzględnia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



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)

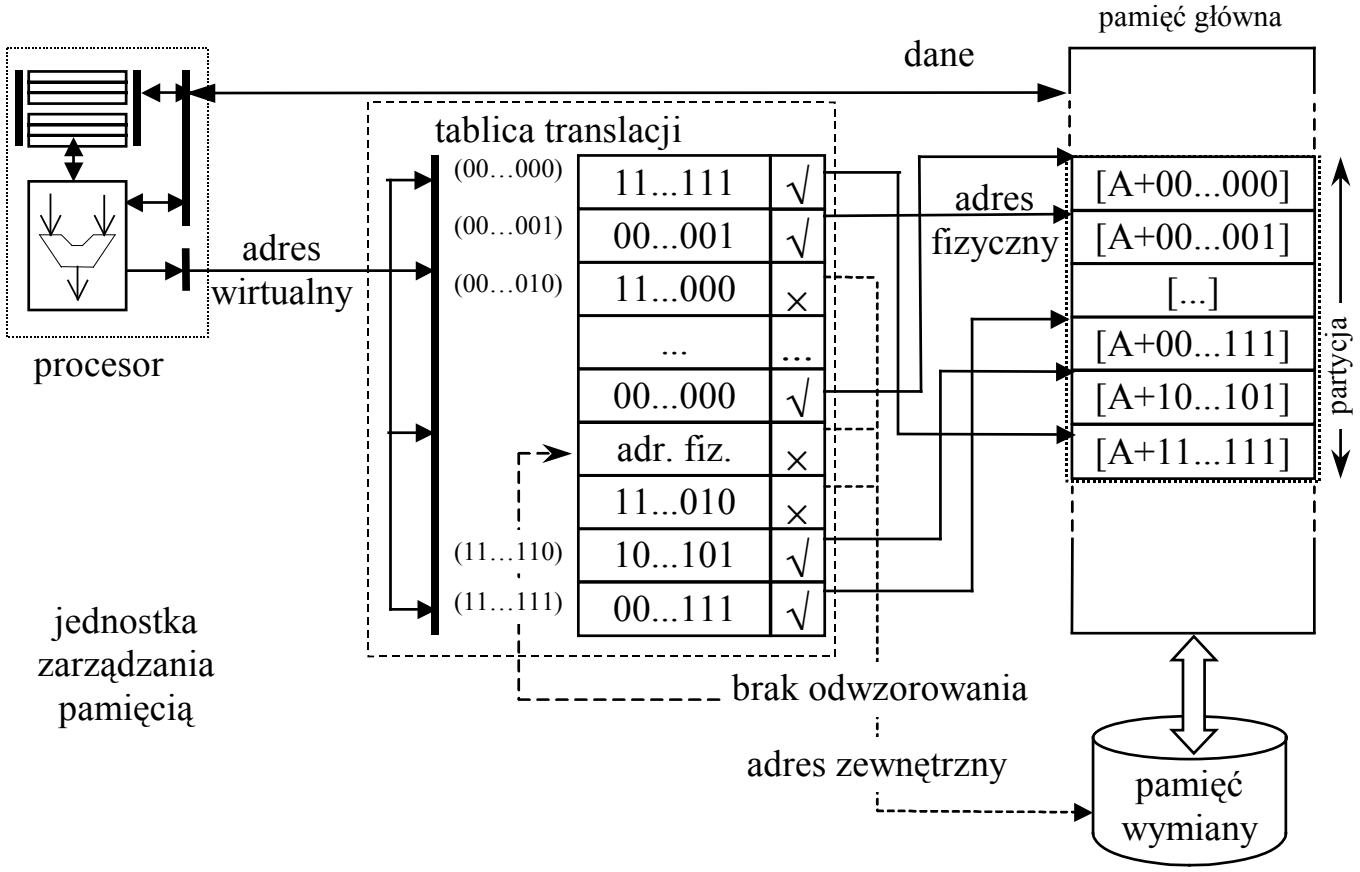
Strategia optymalna (MIN) – wymiana bloku, który będzie użyty najpóźniej
teoretyczna, wymaga antycypacji kolejności wymian.

Strategie wymian dla partycji zmiennych (przy stronicowaniu):

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

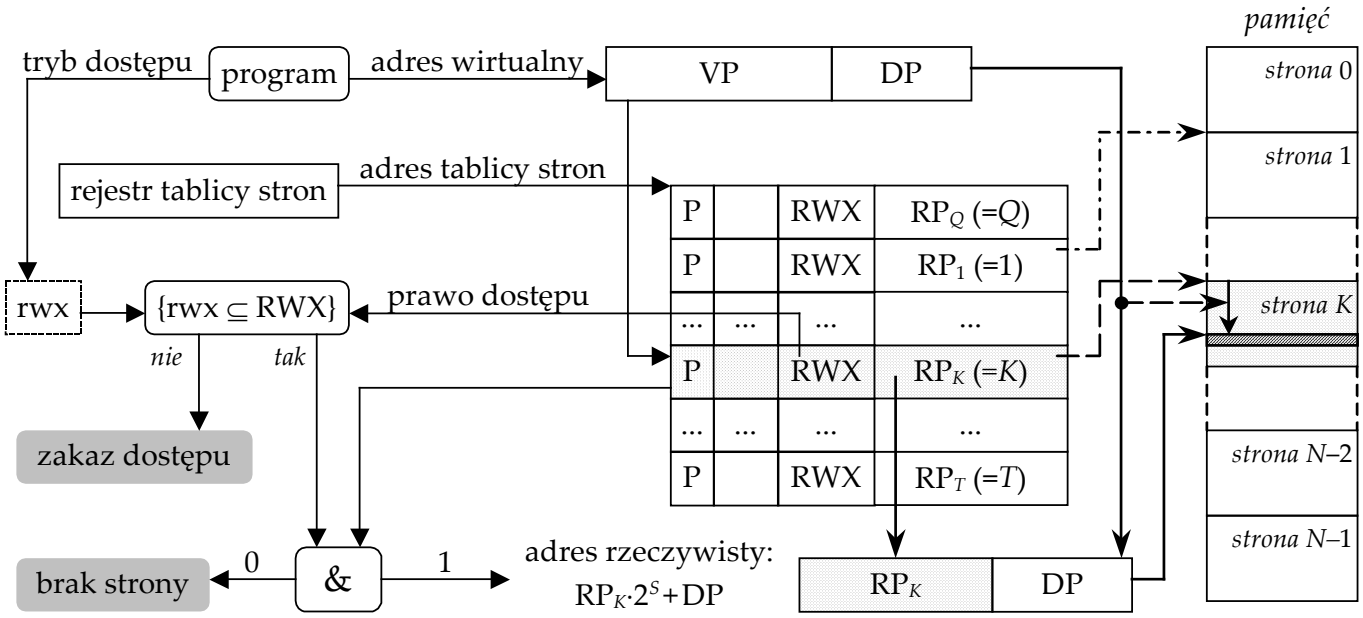
Strategia optymalna VMIN – wymiana bloku, który będzie użyty najpóźniej, lecz z możliwością zmiany rozmiaru partycji.
teoretyczna, wymaga antycypacji kolejności wymian.

Opis odwzorowania – translacja adresu



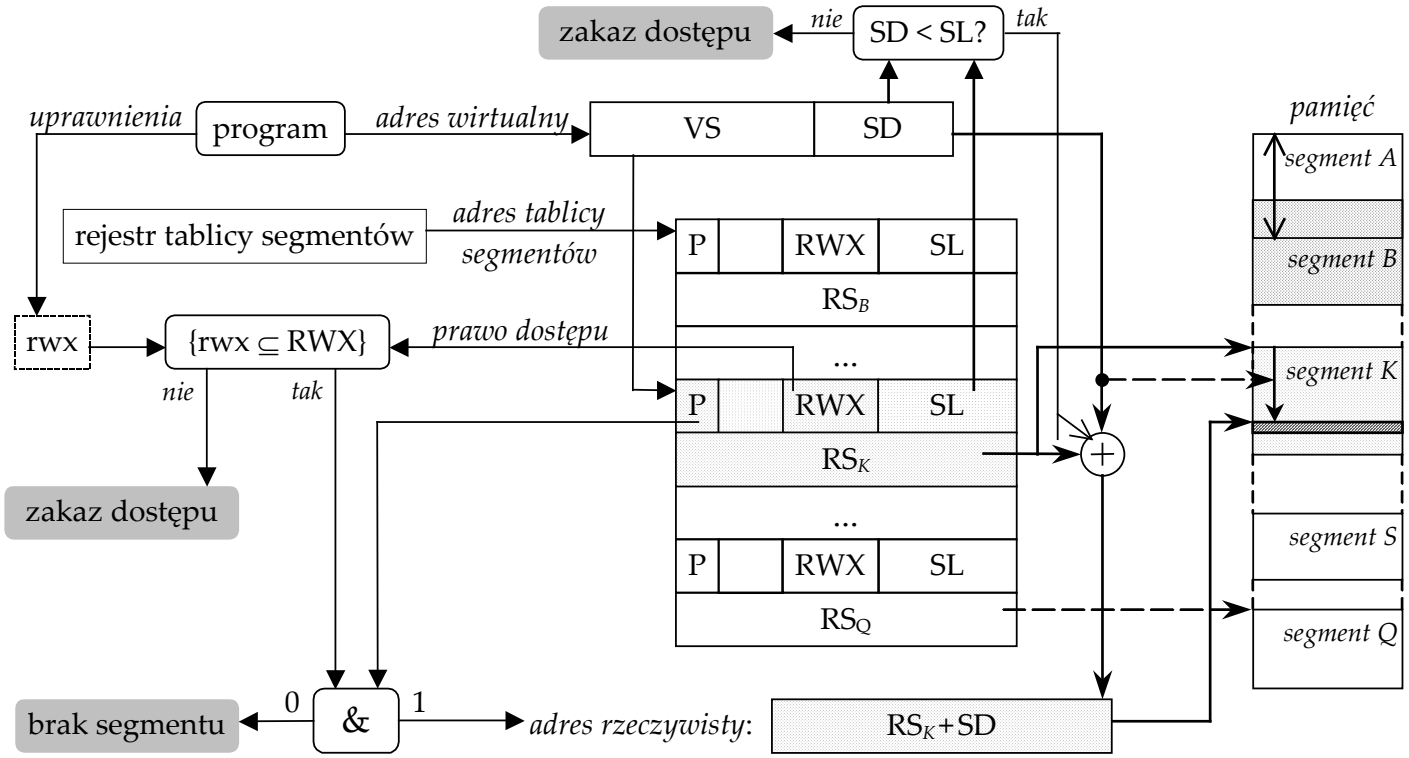
Translacja adresu w układzie zarządzania pamięcią

Stronicowanie



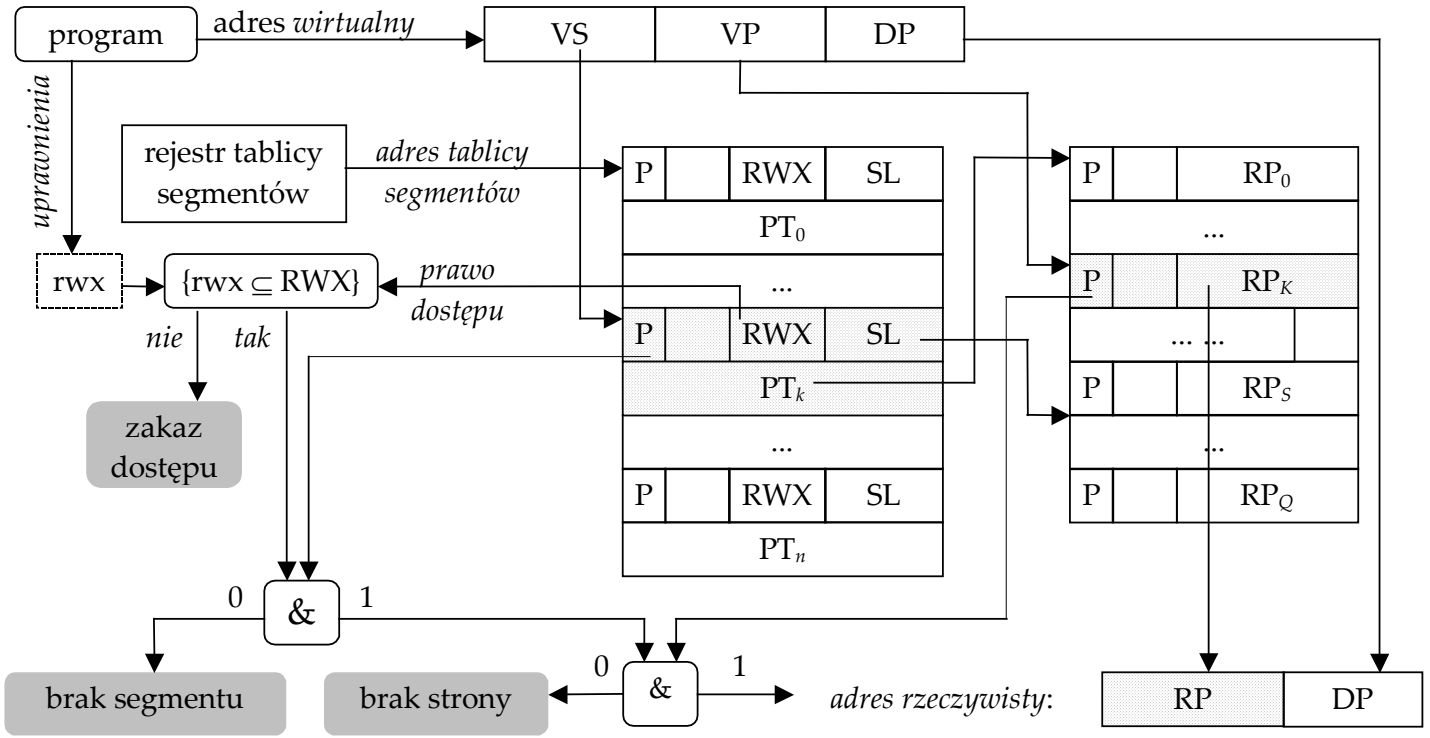
P – bit obecności strony, RWX – kod praw dostępu, VP – numer strony wirtualnej, RP – numer strony rzeczywistej, DP – przemieszczenie na stronie

Segmentacja



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

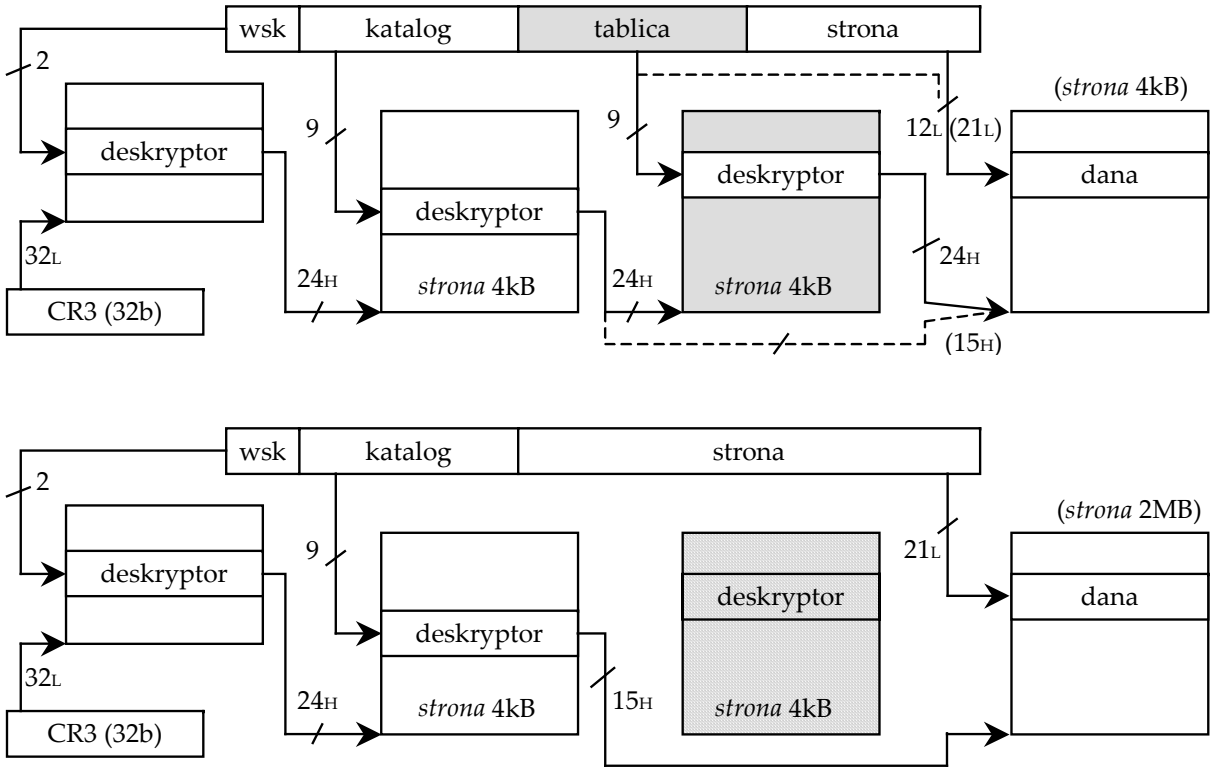
Segmentacja stronicowana



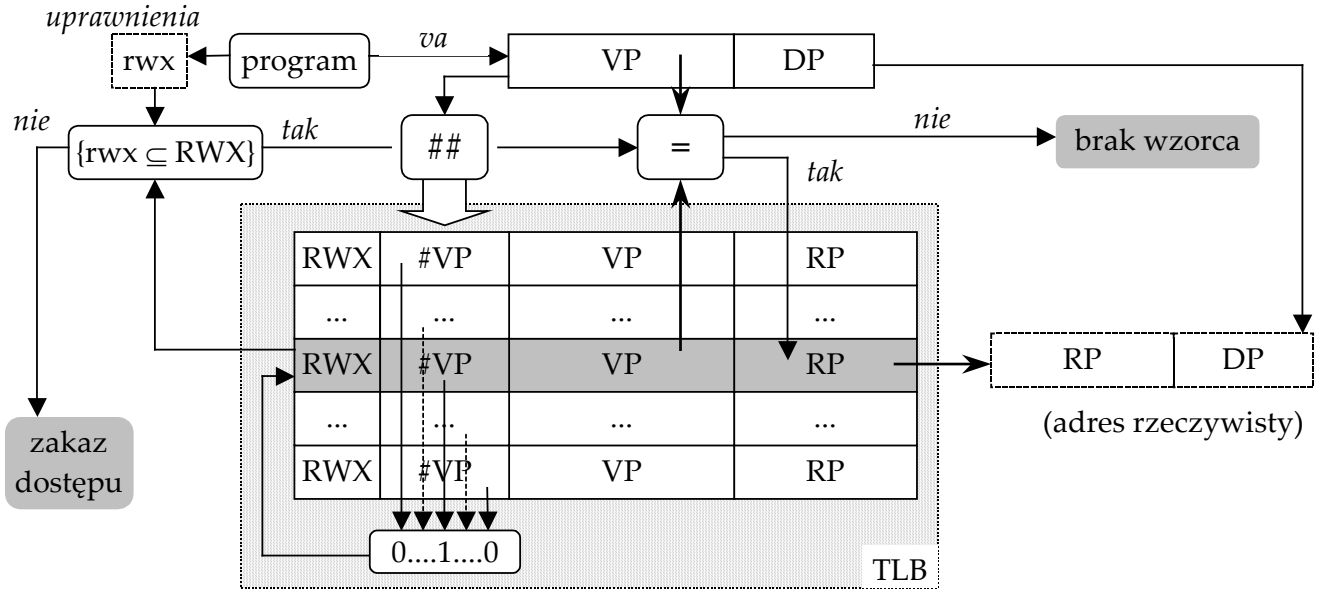
Translacja adresu w trybie segmentacji stronicowanej

Katalog stron

Wielopoziomowa tablica stron (IA-32) – (36-b adres fizyczny)

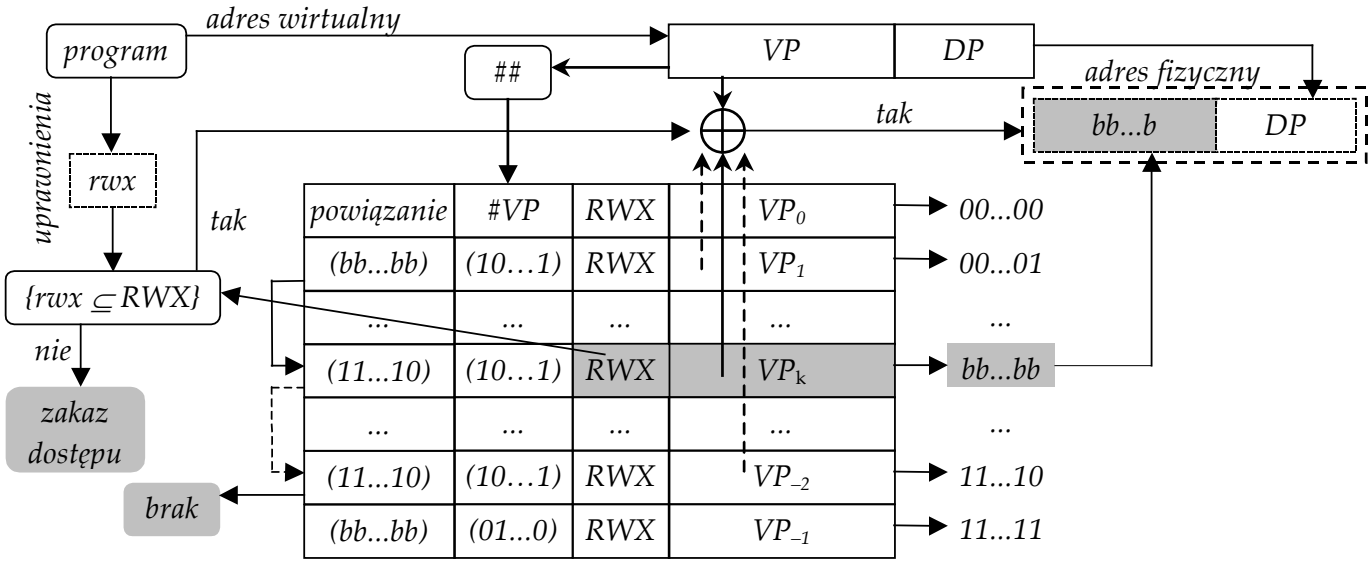


Pamięć podręczna tablicy stron (TLB)



Bufor antycypacji translacji TLB (ang. *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

Przydział pamięci segmentowanej

segment umieszczany w pierwszej dziurze o wystarczającym rozmiarze

Metody – tworzenie i przeszukiwanie list:

- BF – *najlepsze wpasowanie* (ang. *best fit*) – uporządkowanych według rosnących rozmiarów
- WF – *najgorsze wpasowanie* (ang. *worst fit*) – uporządkowanych według malejących rozmiarów
- FF – *pierwsze wpasowanie* (ang. *first fit*) – nieuporządkowanych,
- BB – *wpasowanie binarne* (ang. *binary buddy*) – 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)