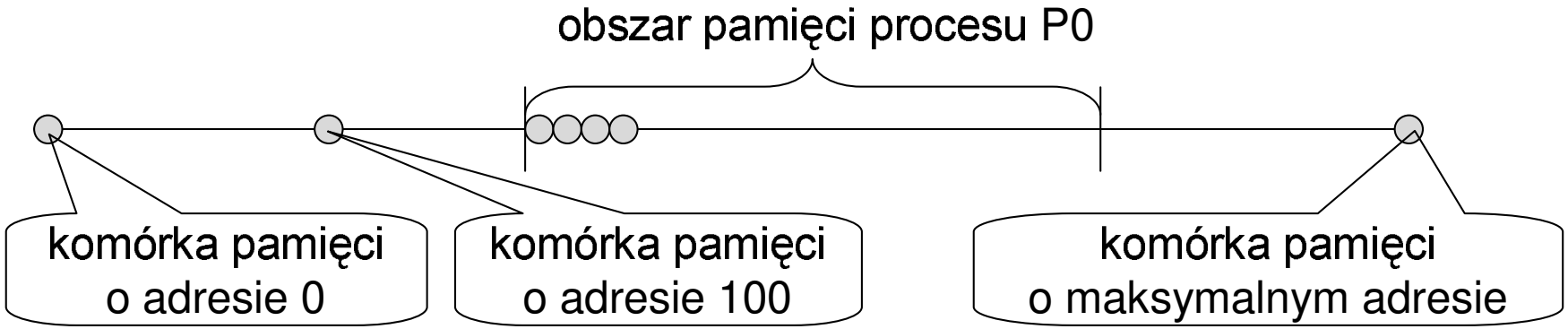
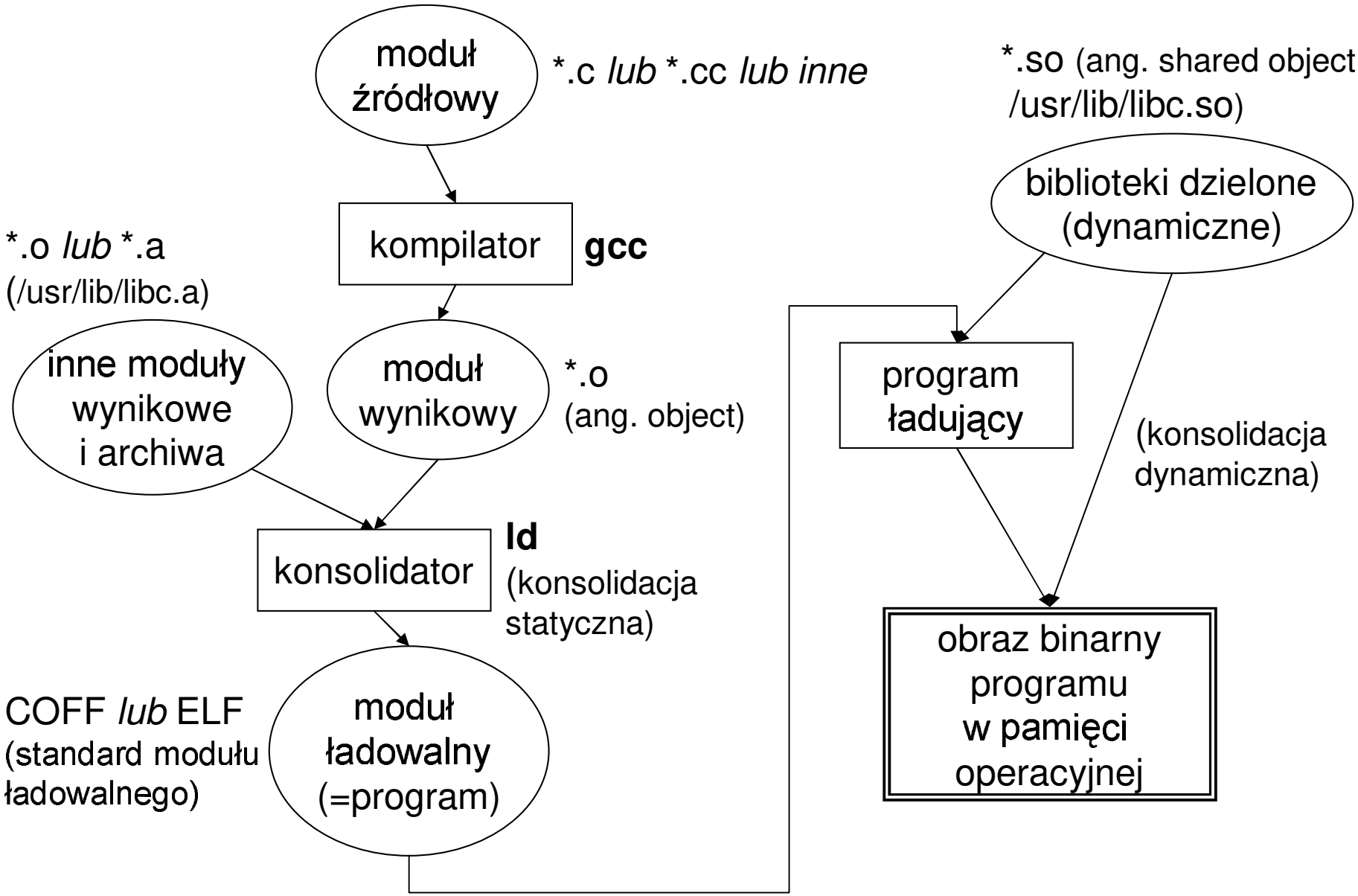
Zarządzanie pamięcią operacyjną

Zarządzanie pamięcią operacyjną;

- każdy proces wymaga *obszaru pamięci operacyjnej* w którym przechowuje *kod* i *dane*
  (*kod* składa się z rozkazów; procesor pobiera je z pamięci operacyjnej, dekoduje i wykonuje – zauważmy że to też wymaga odwołań do pamięci operacyjnej, podobnie jak dostęp do danych ...)
- zarządzanie pamięcią operacyjną komplikuje się w systemach wieloprogramowych, gdyż procesy muszą "współdzielić" pamięć operacyjną między sobą
- *założenie*: na razie pamięć operacyjna traktujemy jako tablicę ponumerowanych komórek pamięci (indeksy tej tablicy to adresy komórek; każda komórka zawiera jeden bajt 0..255)



Zarządzanie pamięcią operacyjną;
fazy tworzenia i uruchamiania programu (Unix)



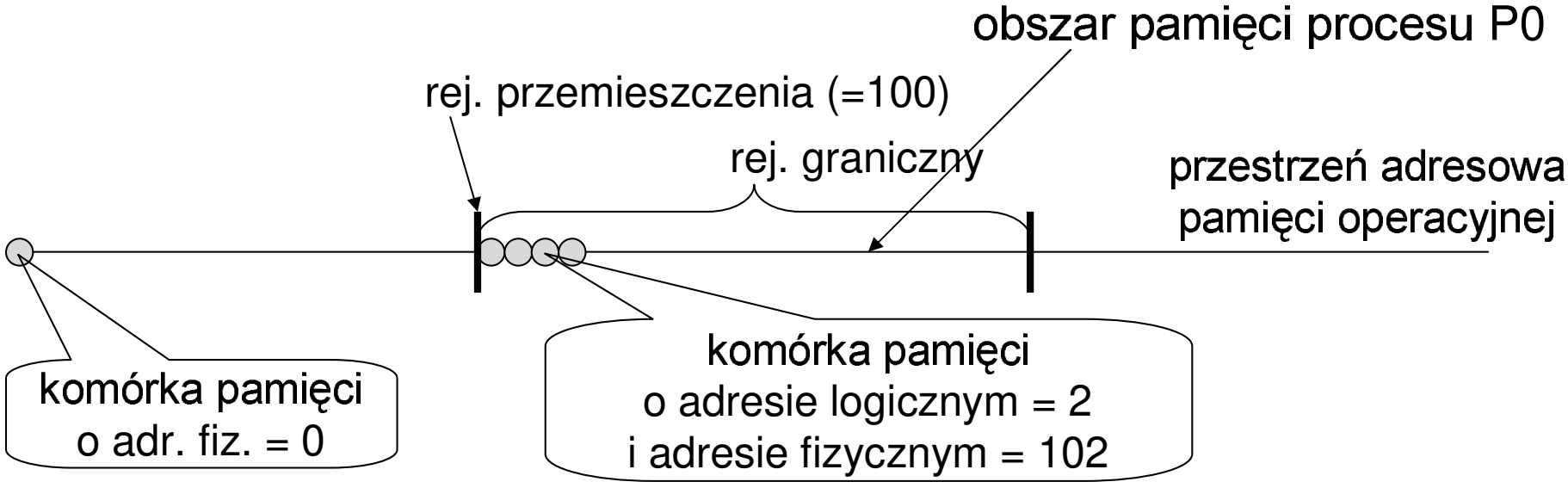
Zarządzanie pamięcią operacyjną; **wiązanie adresów**

- mamy trzy rodzaje "adresów" (procedur i zmiennych w programach):
  1. nazwy symboliczne
     (w modułach źródłowych, np *.c, *.cc)
  2. adresy względne
     (adresy liczone "względem" początku modułu wynikowego)
  3. adresy bezwzględne
     (adresy komórek pamięci operacyjnej)
- "wiązanie adresu" ⇔ zamiana adresu 1 -> 3
- zamianę 1 -> 2 wykonuje kompilator
- zamiana 2 -> 3 może być robiona na różne sposoby:
  - przez konsolidator
    (z góry ustalony adres pod który musi być załadowany program
    – kiepski pomysł ...)
  - w czasie ładowania programu
    (program nie może się przesuwać w pamięci po załadowaniu; DOS, *.exe)
  - w czasie działania programu
    (wymaga *rejestru przemieszczenia* [patrz: następny slajd])

Zarządzanie pamięcią operacyjną; **rejestr przemieszczenia**

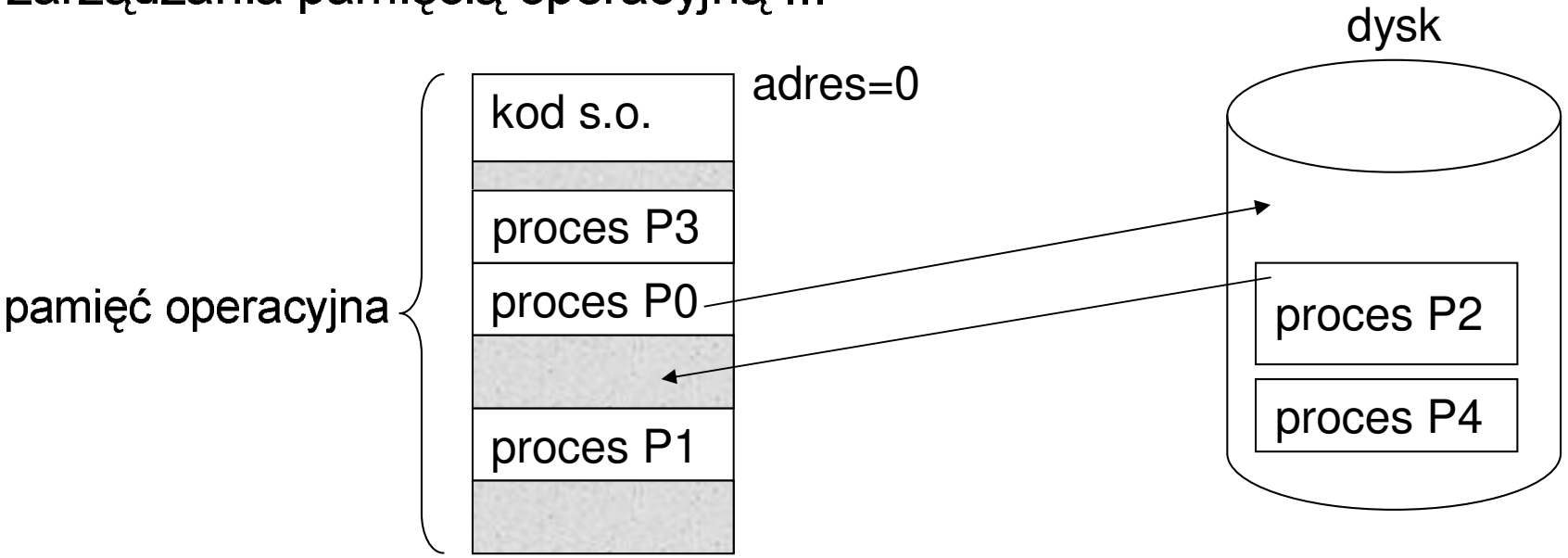
- pojawia się rozróżnienie na adresy komórek pamięci *logiczne* i *fizyczne*
- adres fizyczny = "rej. przemieszczenia" + adres logiczny
- procesor posługuje się wyłącznie adresami logicznymi
- adr. fizyczny ⇔adres bezwzględny, adr. logiczny ⇔adres względny
- rejestr graniczny służy do ochrony pamięci
- podczas przełączania się procesora (od procesu do procesu) zmienia się wartość rejestru przemieszczenia (jest przechowywana w BKP)
- *przykład*: Intel 8086, DS*16, CS*16 są rejestrami przemieszczenia

```
mov AX, [123]
    123 -adr. log.
    DS*16+123 -adr. fiz.
```



Zarządzanie pamięcią operacyjną; **wymiana procesów [ang. swapping]**

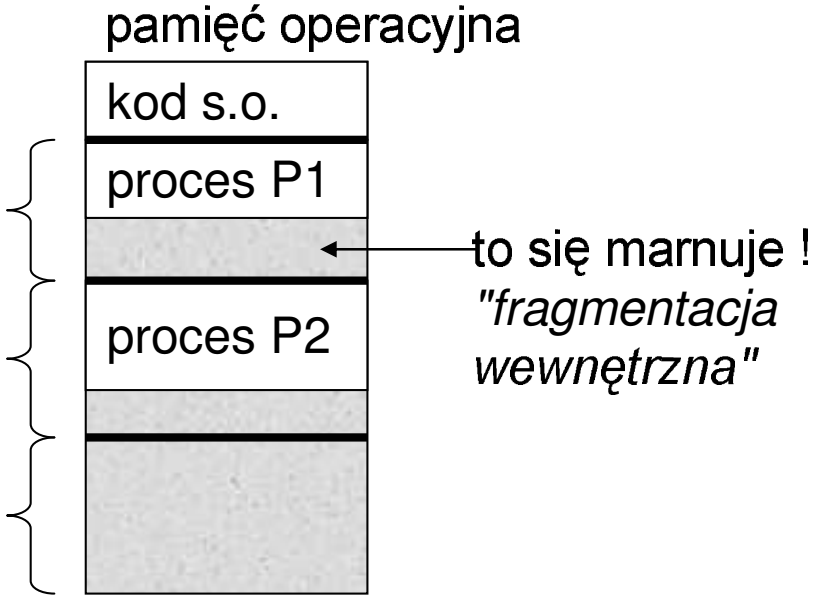
- aby w s.o. mogło działać więcej procesów niż się mieści w pamięci operacyjnej niektóre z nich (tymczasowo) zapisuje się na dysku ...
- decydowaniem który proces "wymienić" zajmuje się planista średnio-terminowy
- w wyniku wymiany i/lub kończenia się procesów w pamięci operacyjnej powstają niewykorzystane obszary (tzw. "dziury w pamięci") – gdy sprowadzamy proces z dysku do pamięci to może on trafić do takiego obszaru; widać że proces może trafiać pod różne adresy – stąd niezbędne jest zastosowanie (przynajmniej) rej. przemieszczenia do zarządzania pamięcią operacyjną !!!



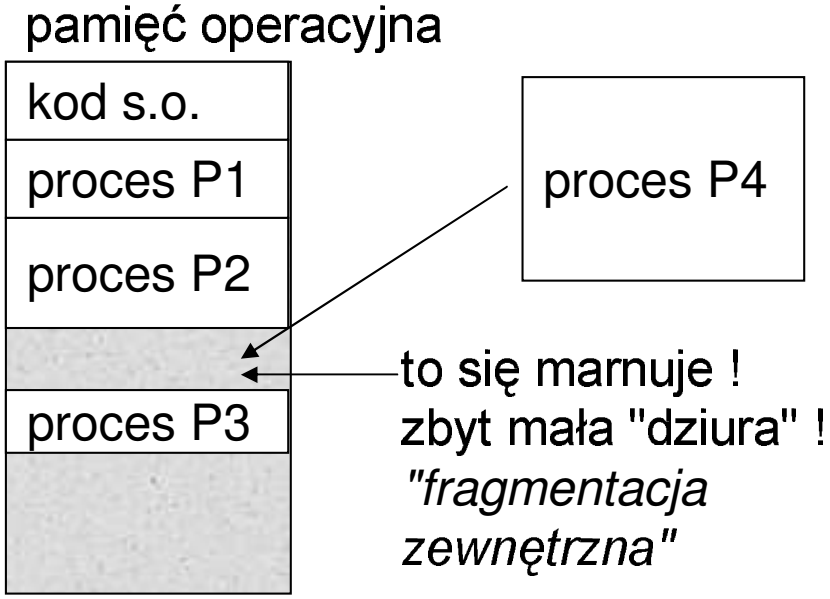
Zarządzanie pamięcią operacyjną; **sposoby przydzielania obszarów p.o. procesom**

- dwie metody:
  - obszary stałej długości (metoda przestarzała ...)
  - obszary zmiennej długości
    - strategie przydziału "dziury w pamięci" procesowi (aby uniknąć fragm. zewn.)
      - pierwsze dopasowanie (pierwsza dostatecznie duża dziura)
      - najlepsze dopasowanie (najmniejsza z dziur dostatecznie dużych)
      - najgorsze dopasowanie

Obszary stałej długości



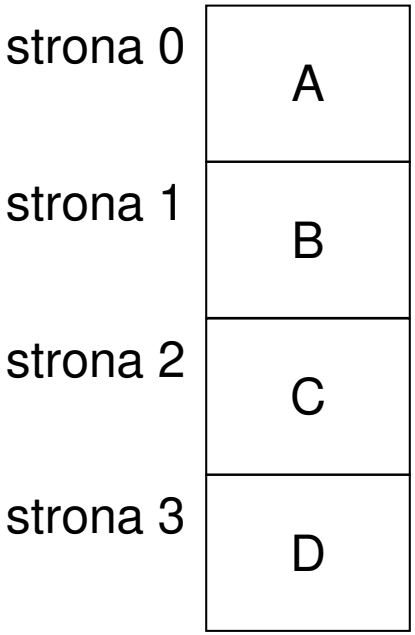
Obszary zmiennej długości



Zarządzanie pamięcią operacyjną; **stronicowanie**

- rozwiązuje problem fragmentacji zewnętrznej ...
- podobnie jak rej. przes. jest to rozwiązanie wymagające wsparcia sprzętowego (w procesorach Intela pojawiło się dopiero w 80386)
- pojęcia: strona pamięci, ramka pamięci, tablica stron

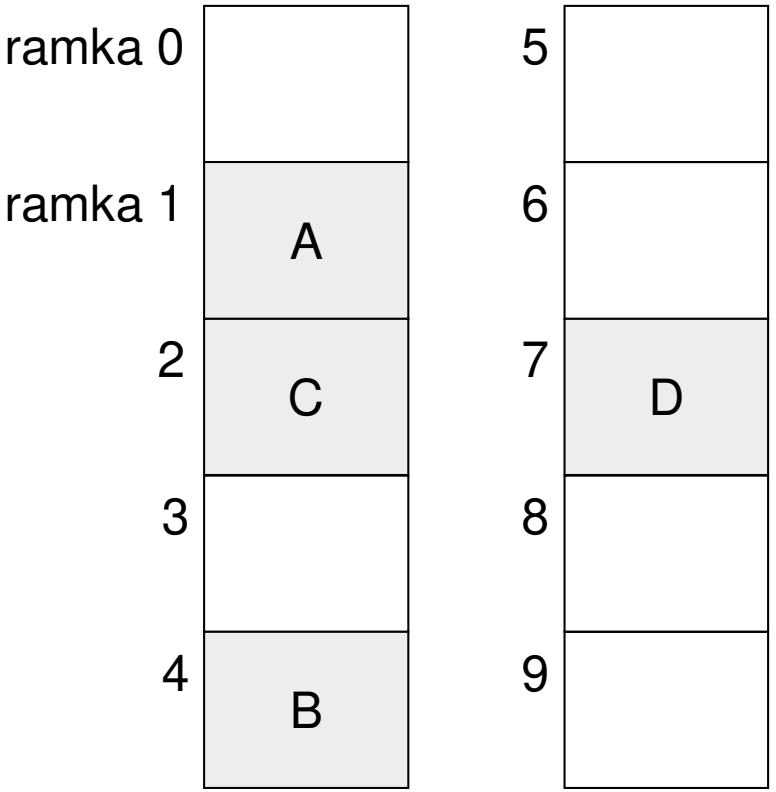
pamięć logiczna procesu
(wirtualna prz. adresowa procesu)
podzielona na strony rozmiaru 4KB



tablica stron

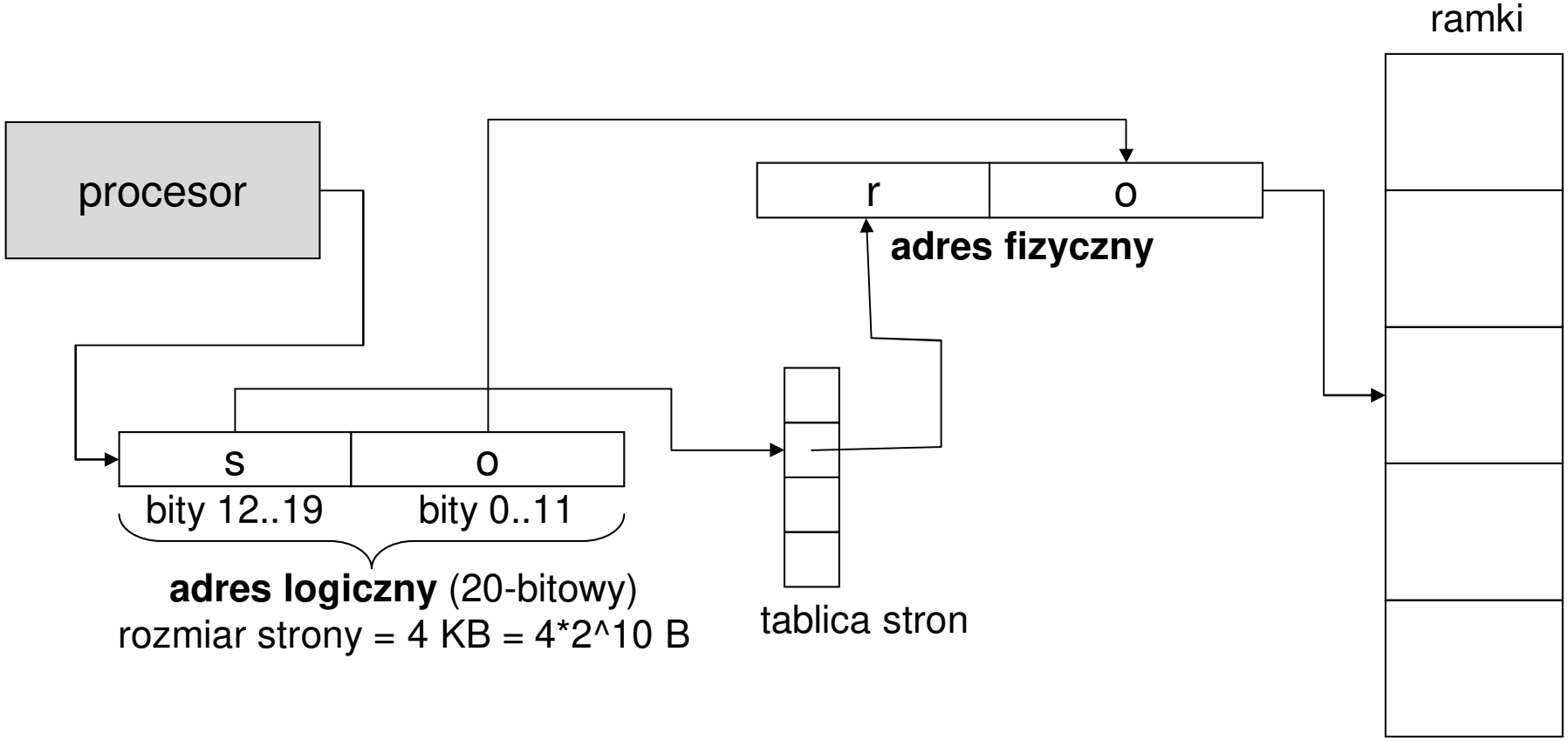
0	1
1	4
2	2
3	7

pamięć fizyczna komputera
podzielona na ramki rozmiaru 4KB



Zarządzanie pamięcią operacyjną; **stronicowanie**

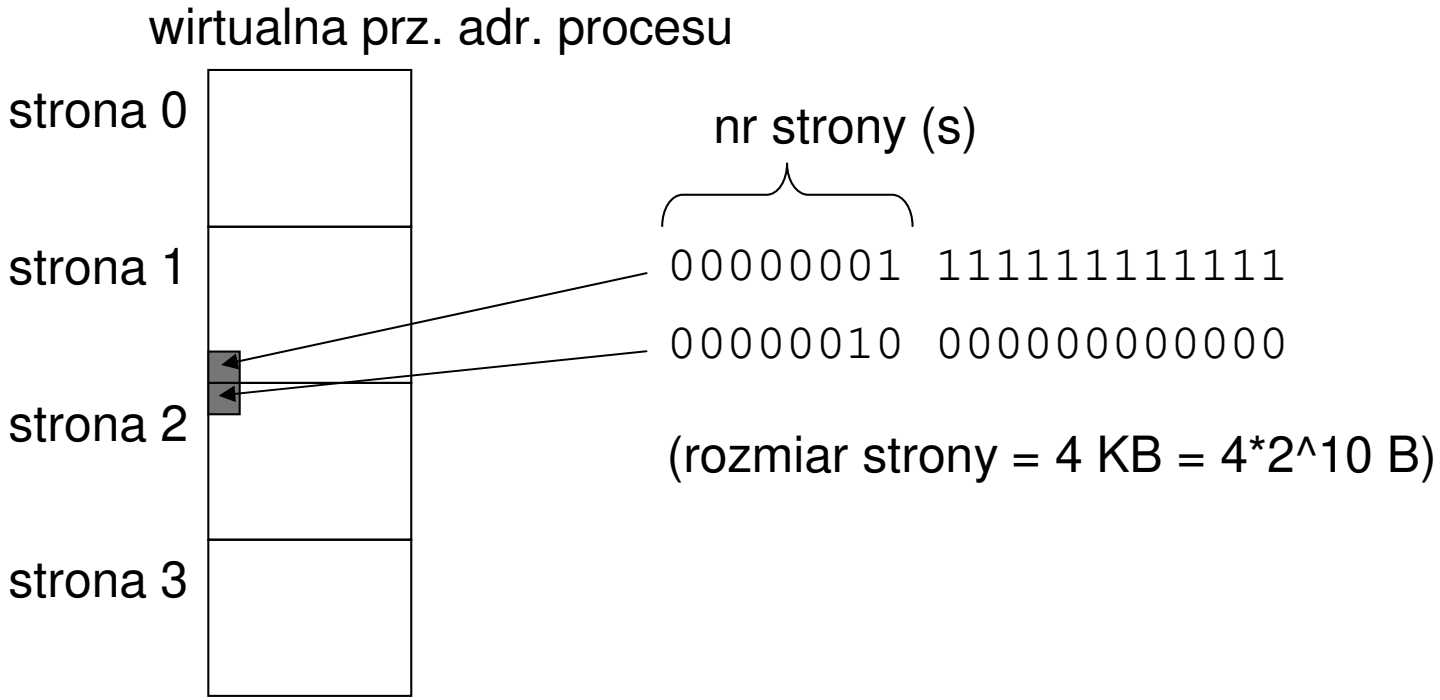
- zamiana adresu logicznego na fizyczny ...
  - adr. log.: s = nr strony, o = przesunięcie wzg. początku strony
  - adr. fiz.: r = nr ramki, o = przesunięcie wzg. początku ramki



... jak duża może być tutaj wirtualna prz. adr. procesu ?

Zarządzanie pamięcią operacyjną; **stronicowanie**

- jak wygląda adres logiczny przy stronicowaniu ?
  - jeśli rozmiar strony (= rozmiarowi ramki) jest potęgą liczby 2 to adres logiczny jest liczbą całkowitą składającą się z:
    - części zawierającej nr strony
    - części zawierającej przesunięcie względem początku strony
  - adresy kolejnych komórek są kolejnymi liczbami całkowitymi !



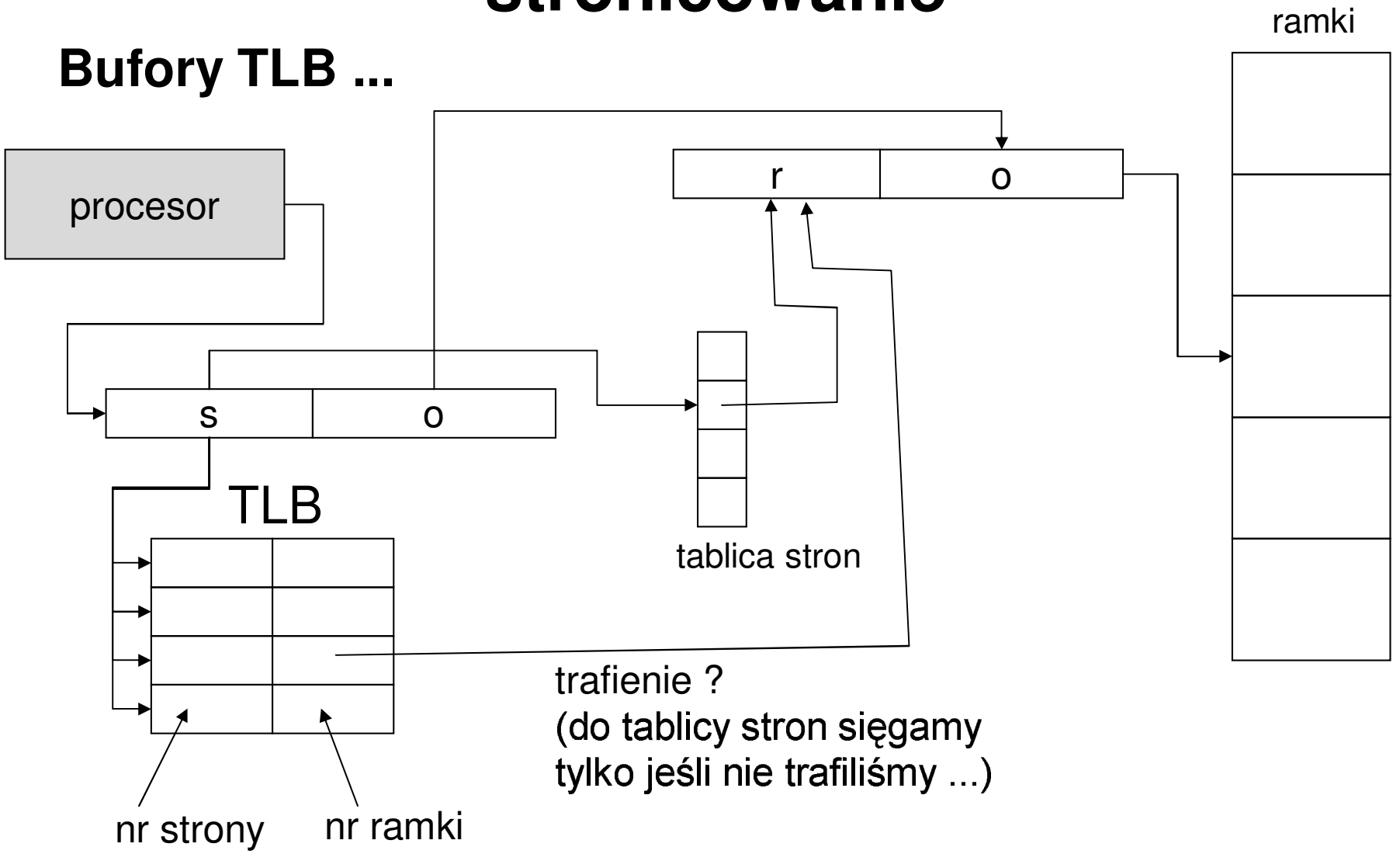
Zarządzanie pamięcią operacyjną; **stronicowanie**

- dlaczego stronicowanie rozwiązuje problem fragmentacji zewnętrznej ?
  - obszar pamięci procesu nie składa się z jednego "dużego kawałka" lecz z wielu "małych kawałków" (=ramek)
  - jeśli tylko jest dostateczna ilość wolnych ramek w pamięci fizycznej to można przydzielić pamięć procesowi
  - stronicowanie nie likwiduje fragmentacji wewnętrznej – ostatnia ramka może być nie w pełni wykorzystywana ...
- stronicowanie spowalnia dostęp do pamięci operacyjnej
  - wymaga dodatkowego dostępu do pamięci aby odczytać tablicę stron
  - *rozwiązanie*: bufory TLB (Translation Look-aside Buffers)
    [patrz: następny slajd]
  - procesor Intel 80486 ma ich 32 ...

Zarządzanie pamięcią operacyjną;

stronicowanie

Bufory TLB ...



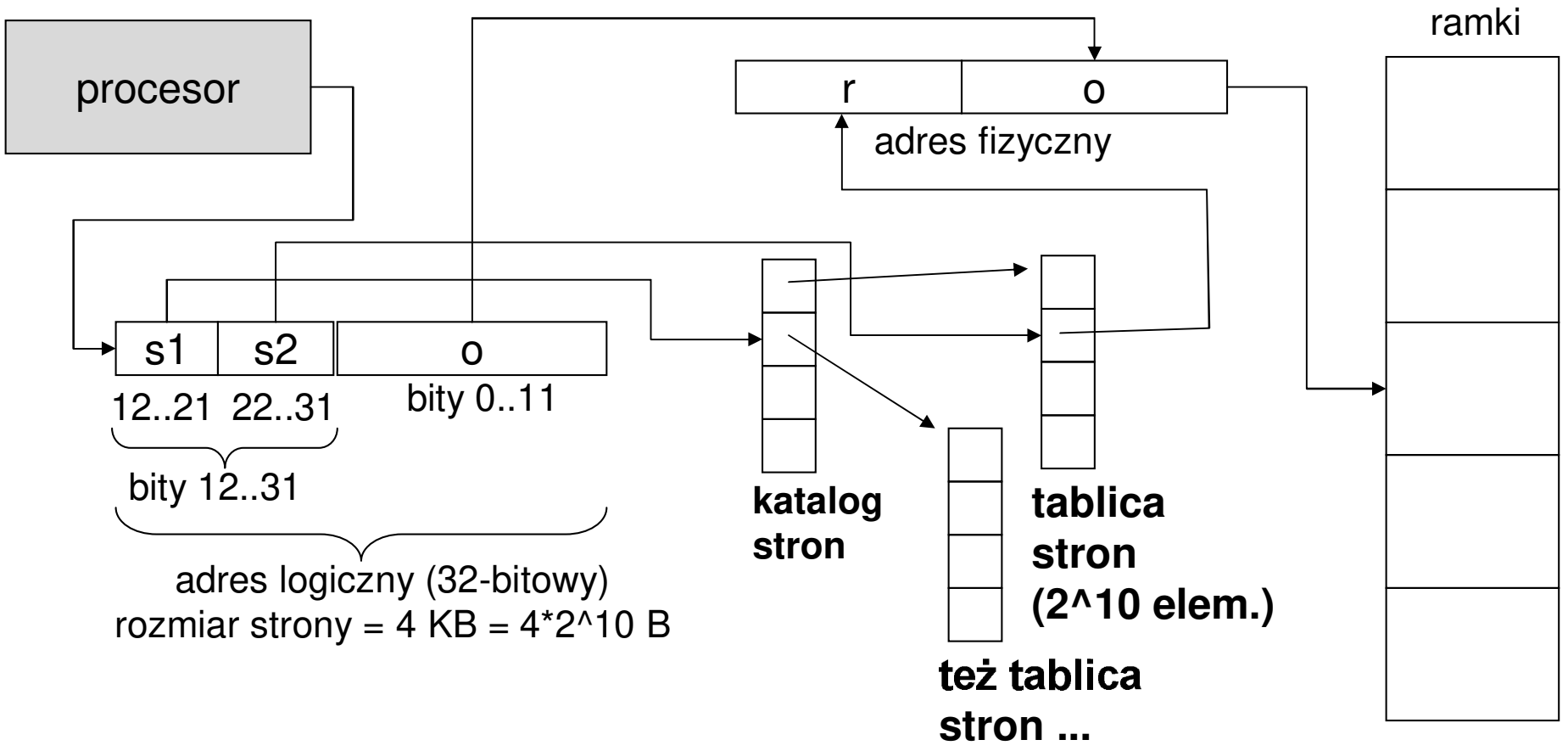
TLB to "pamięć asocjacyjna"; działa bardzo szybko - wszystkie elementy tablicy równocześnie sprawdzają czy przypadkiem nie zawierają "s" ...

Zarządzanie pamięcią operacyjną; **stronicowanie**

- stronicowanie wielopoziomowe
  - gdy tablica stron jest zbyt duża (np. nr strony zapisywany na 20 bitach);
    pamiętajmy ze każdy proces ma własna tablicę stron !!!
  - "katalog stron" i "tablica stron"; element katalogu wskazuje na tablice stron, ale nie każdy element katalogu jest używany ... (?)
  - każdy proces ma własny:
    - "katalog stron"
    - kilka "tablic stron" (tyle ile potrzebuje !!!)
  - chodzi o to aby nie marnować miejsca w pamięci operacyjnej na bardzo dużą tablice stron (która nie jest wykorzystywana ...)

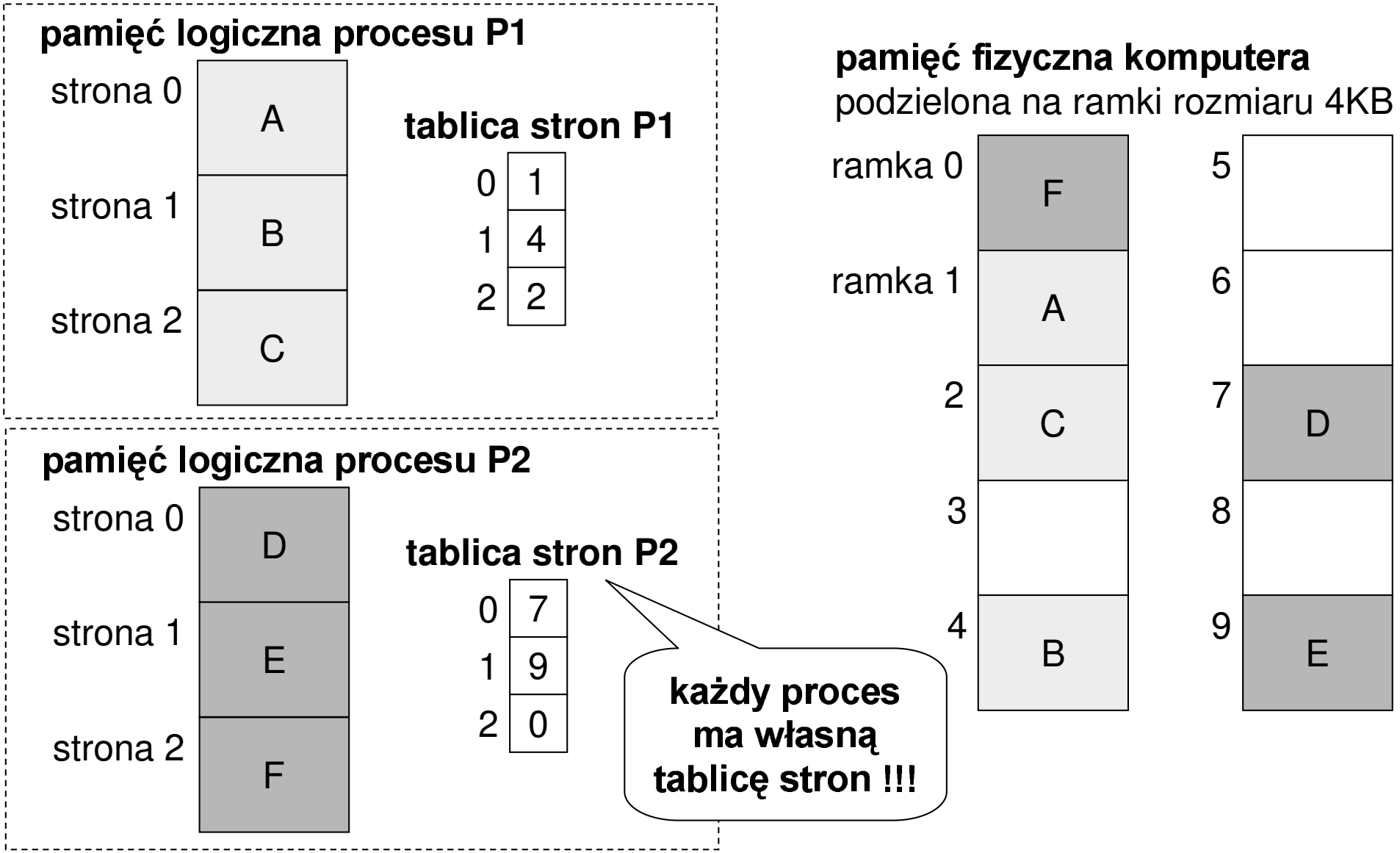
Zarządzanie pamięcią operacyjną; **stronicowanie**

- stronicowanie wielopoziomowe c.d.



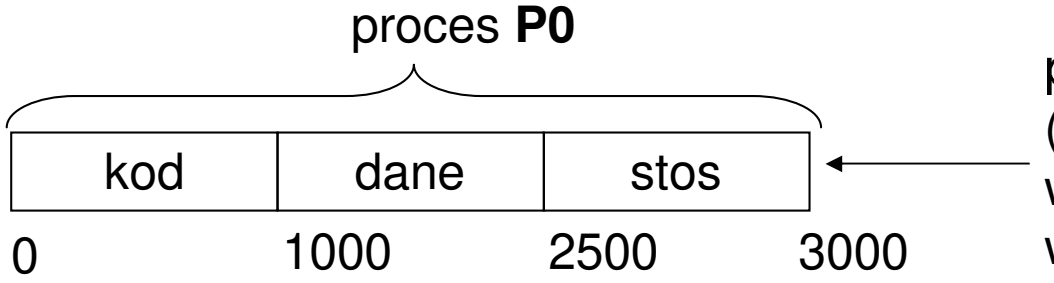
Zarządzanie pamięcią operacyjną; **stronicowanie**

- stronicowanie z pkt. widzenia dwóch procesów ...



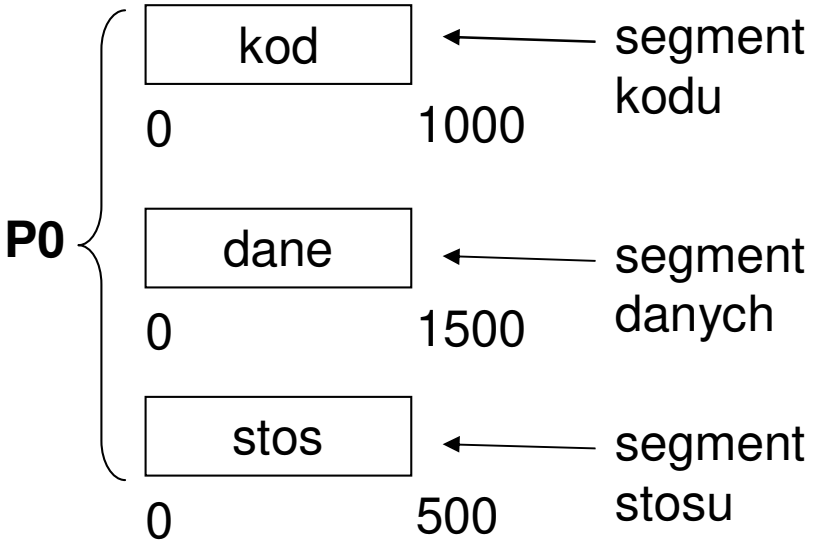
Zarządzanie pamięcią operacyjną; **segmentacja**

- procesy nie potrzebują jednej, spójnej przestrzeni adresowej ...



pamięć logiczna procesu **P0**
(= wirtualna prz. adr. procesu **P0**)
wszystkie *sekcje* programu
w jednym *segmencie* ...

- każdą *sekcje* programu można umieścić w innym *segmencie ...*



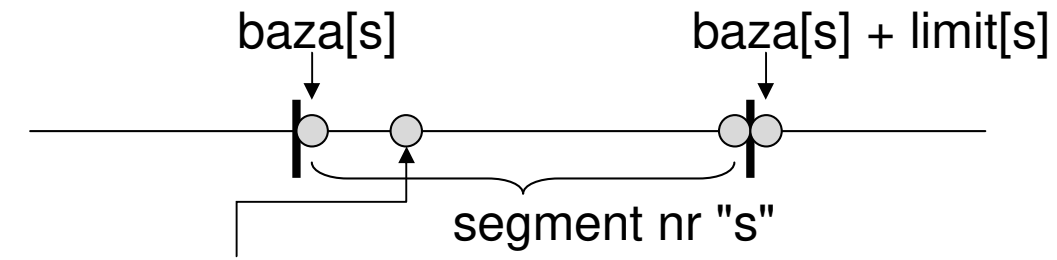
- zalety segmentów:
  - inne prawa dostępu do segmentów o różnym przeznaczeniu (np. kod -tylko odczyt, dane -zapis i odczyt; nieprawidłowy dostęp do pamięci spowoduje pułapkę !)
  - dzielenie danego segmentu przez kilka procesów (umożliwia realizacje bibliotek dzielonych !)

Zarządzanie pamięcią operacyjną; **segmentacja**

- adresy logiczne przy segmentacji:

  s : o

  gdzie s = nr segmentu, o = przesunięcie [ang. offset]
- adresy NIE są tutaj kolejnymi liczbami całkowitymi ...
- *segment* jest opisany przez *bazę* i *limit* (=długość segm.) w tablicy segmentów (zakładamy tutaj że pamięć fizyczna jest *tablicą komórek pamięci*)

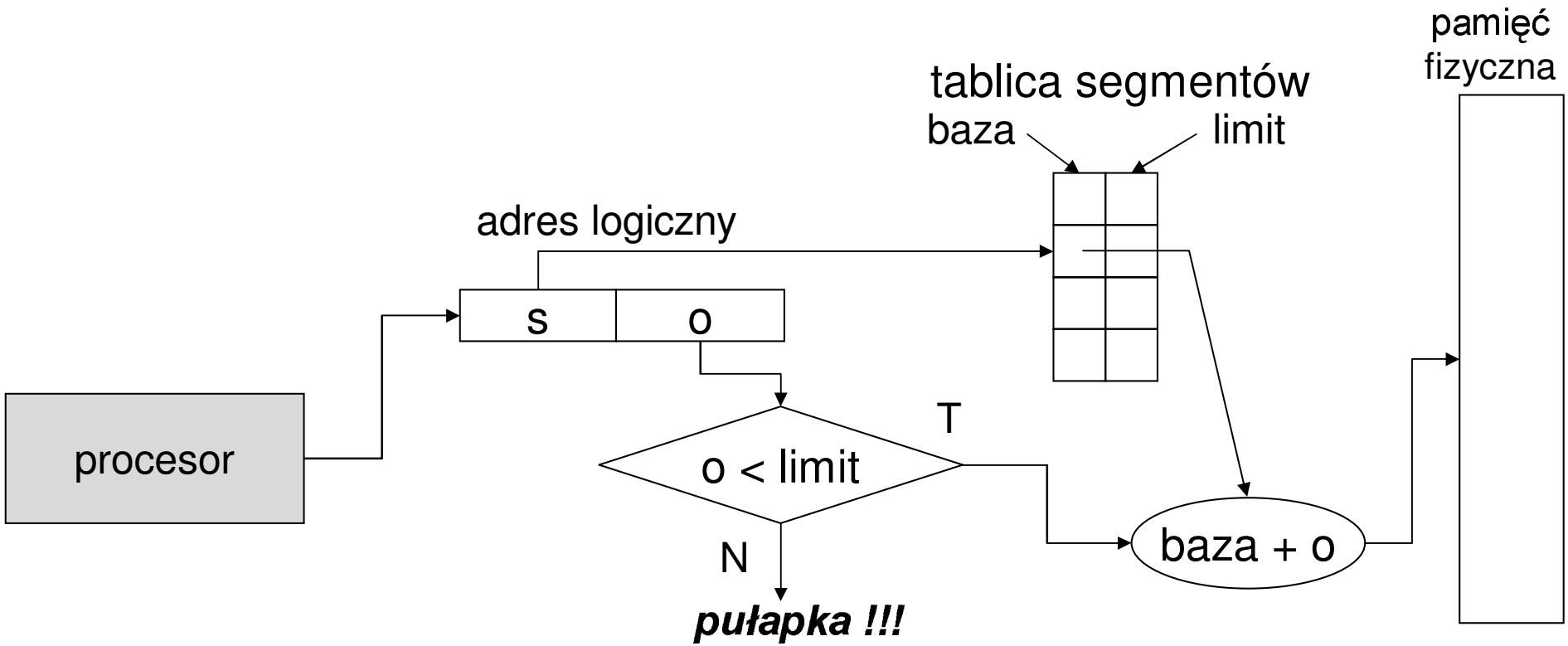


adr. log = s **:** o

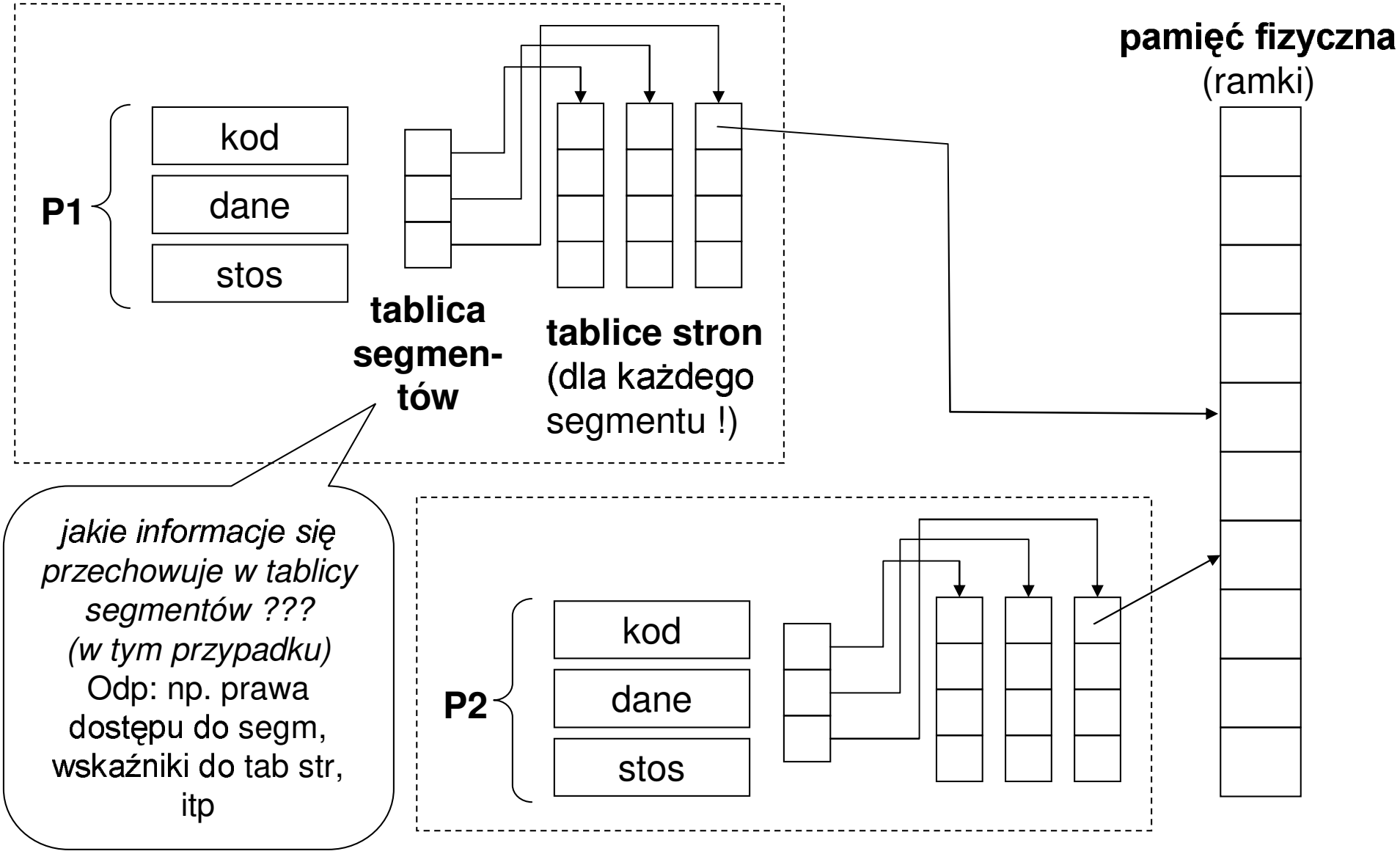
adr. fiz = baza[s] + o

Zarządzanie pamięcią operacyjną; **segmentacja**

- zamiana adresu logicznego na fizyczny przy segmentacji ...



Zarządzanie pamięcią operacyjną; **współczesne rozwiązanie: segmentacja + stronicowanie**



Zarządzanie pamięcią operacyjną; **zarządzanie pamięcią w 80386**

- rodzina procesorów Intel-a:
  - 8086; rej. przemieszczenia; CS, DS, SS
  - 80286; 2 tryby pracy
    - real (jak 8086)
    - protected (segmentacja !)
  - 80386; 3 tryby pracy
    - real (jak 8086)

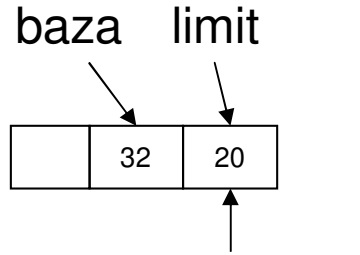
    - protected
    - V86 (maszyny wirtualne 8086)

    } segmentacja + stronicowanie !

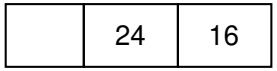
Zarządzanie pamięcią operacyjną; **zarządzanie pamięcią w 80386**

- używamu tutaj terminologii Intel-a !!!
- tablica deskryptorów (=*tablica segmentów*)
- deskryptor = opisuje jeden segment
- selektor = indeks do tablicy deskryptorów
  - tablica deskr. ma 2^13 elem
- GDT = globalna tablica deskryptorów
  - istnieje jedna GDT
  - opisuje segmenty dostępne(?) dla wszystkich procesów
- LDT = lokalna tablica deskryptorów
  - każdy proces ma własną LDT
  - opisuje segmenty dostępne tylko dla danego procesu
  - LDT poszczególnych procesów są segmentami GDT
- GDTR = rejestr wskazujący GDT (GDTR zawiera adr. fizyczny tablicy GDT)
- LDTR = rejestr wskazujący bieżącą LDT (LDTR zawiera selektor do GDT)

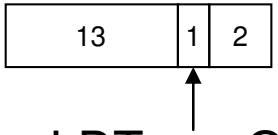
deskryptor/386 :



deskryptor/286 :



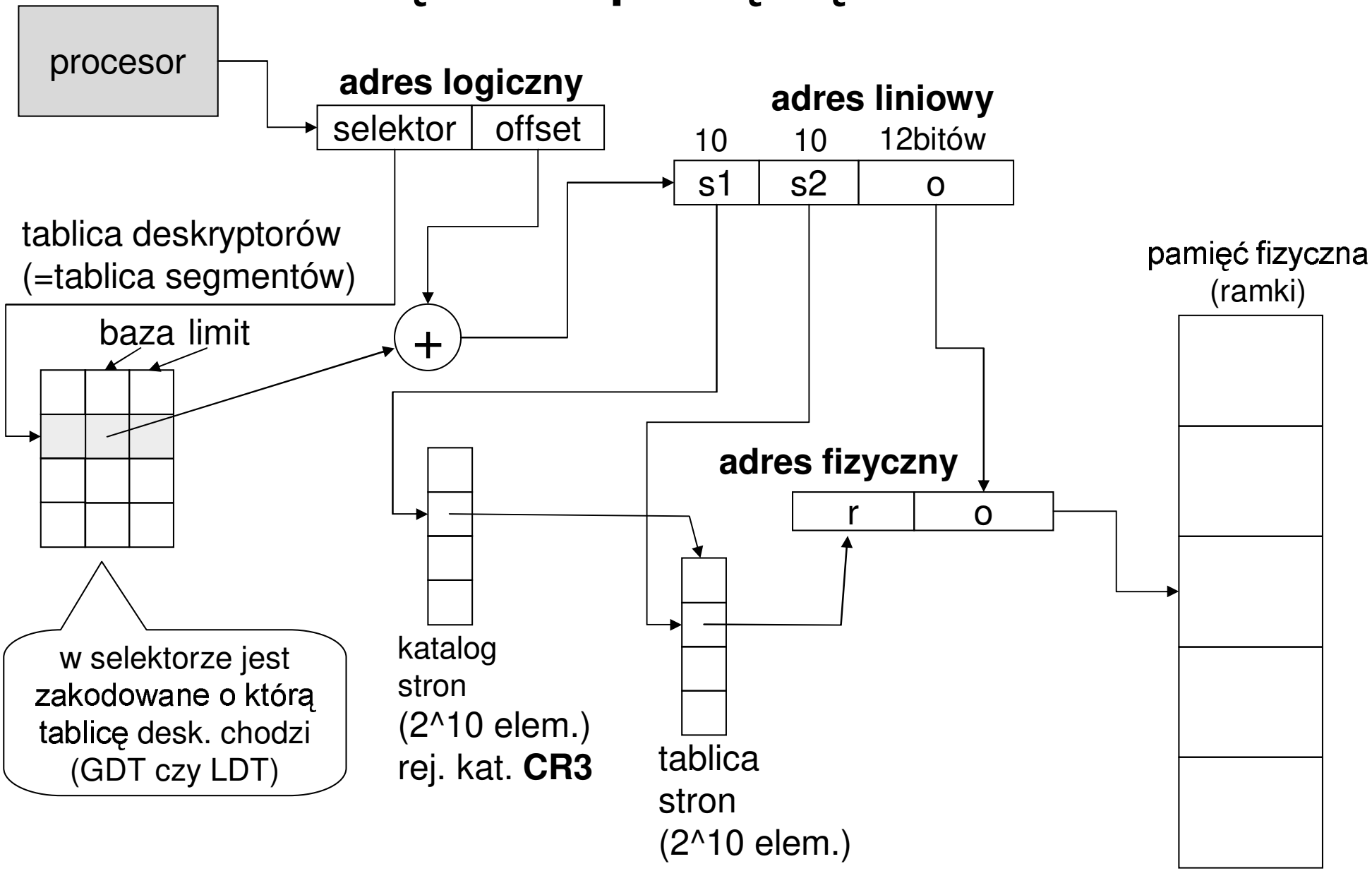
selektor :



Zarządzanie pamięcią operacyjną; **zarządzanie pamięcią w 80386**

- adres logiczny= selektor **:** offset
  selektor= 16bitów; offset= 32bity
- przestrzenie adresowe:
  - **logiczna**
    - adresy logiczne "selektor : offset"
  - **liniowa**
    - w niej "leżą" segmenty opisywane przez deskryptory
    - *podlega stronicowaniu w 80386 !!! (czyli jest podzielona na "strony" które mają przyporządkowane "ramki" pamięci fizycznej)*
    - w 80286 : pamięć liniowa = pamięć fizyczna
  - **fizyczna**
    - adresy 32bitowe w 80386
    - adresy 24bitowe w 80286
- w procesorze 80386 mamy segmentacje + stronicowanie, lecz istnieje tylko jedna tablica stron (całą prz. adr. liniowa podlega stronicowaniu, a nie poszczególne segmenty ...)
- czy nie ma problemu "fragmentacji zewnętrznej" skoro w pamięci liniowej mogą się pojawić "zbyt małe dziury" ???

Zarządzanie pamięcią operacyjną;
zarządzanie pamięcią w 80386

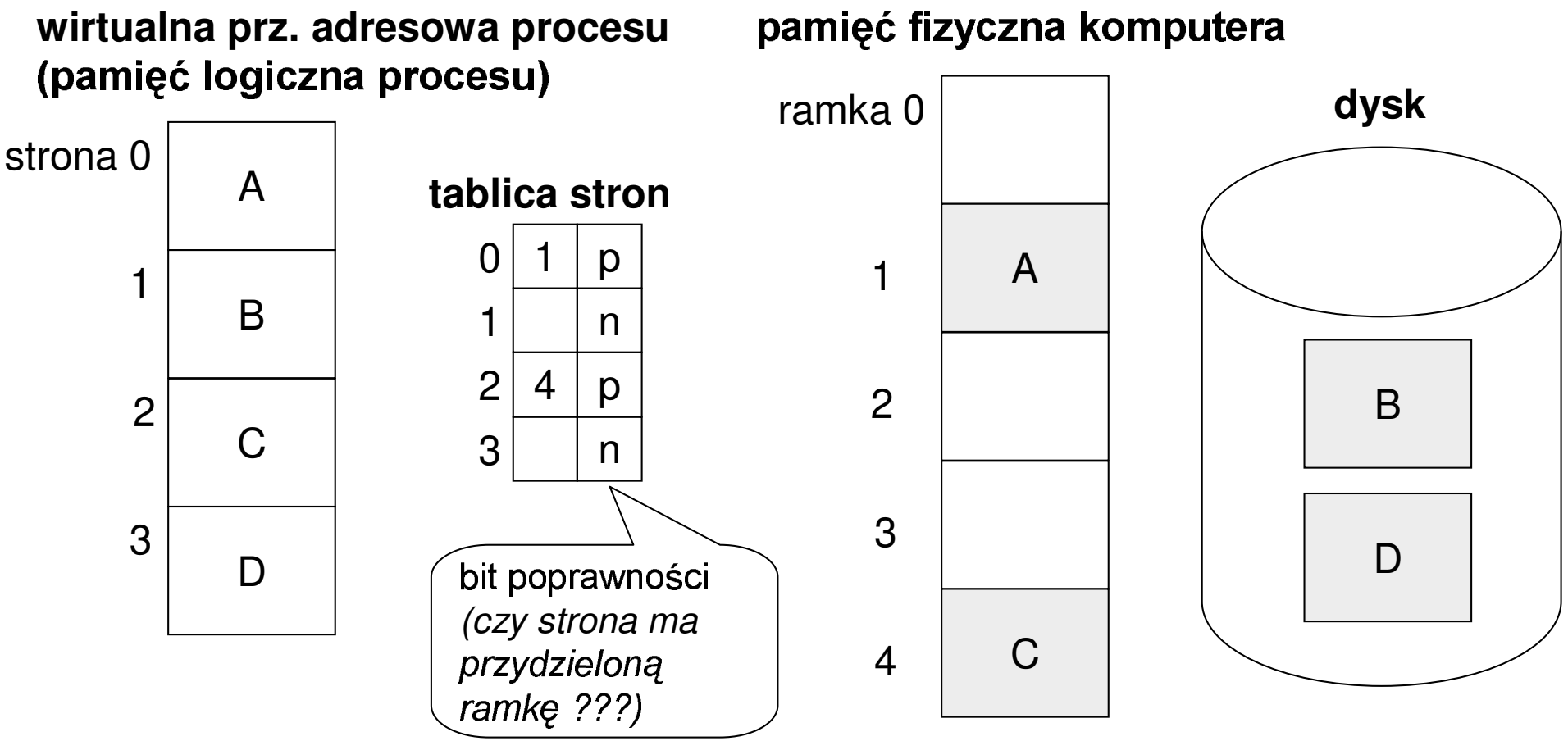


Zarządzanie pamięcią operacyjną; **pamięć wirtualna**

- poprzednio zakładaliśmy że aby proces mógł działać musi się "w całości" znajdować w pamięci operacyjnej...; w wieloprogramowych s.o. jeśli zaczynało brakować pamięci operacyjnej wtedy stosowano "wymianę procesów" (swapping)
- *pamięć wirtualna* to technika pozwalająca na wykonywanie procesów które nie znajdują się "w całości" w pamięci operacyjnej
  - łączna objętość pamięci wykonywanych równocześnie procesów może przekraczać objętość dostępnej pamięci fizycznej !
  - także pamięć pojedynczego procesu może być większa niż dostępna pamięć fizyczna
- pamięć wirtualną realizuje się najczęściej jako "stronicowanie na żądanie" [ang. (on) demand paging]

Zarządzanie pamięcią operacyjną;
pamięć wirtualna

- *zasada*: jeśli proces odwołuje się do *strony* która nie ma przydzielonej *ramki* to jest generowana pułapka "*błąd strony*" [ang. page fault]
- procedura obsługi błędu strony powinna przydzielić ramkę stronie, skopiować zawartość strony z dysku do ramki i wznowić proces ...



Zarządzanie pamięcią operacyjną;
pamięć wirtualna

- procedura obsługi błędu strony (b. szczegółowo):
  - przydziel stronie wolną ramkę
  - skopiuj zawartość strony z dysku do ramki
  - wznów wykonywanie procesu od rozkazu który spowodował błąd stron (jakie to mogą być rozkazy? mov, jmp, call itp)
- co zrobić jeśli nie ma wolnej ramki ???
  - trzeba wybrać "ramkę ofiarę"
  - zapisać jej zawartość na dysku
  - zwolnioną ramkę przypisać stronie która spowodowała błąd strony
- uwagi:
  - ramka ofiara może należeć do innego procesu niż ten który spowodował błąd strony
  - przestrzeń dyskowa do zapamiętywania stron to często osobna partycja dysku magnetycznego (lub specjalny plik w systemie plików użytkownika)
  - kod programu także podlega stronicowaniu na żądanie, ale w tym wypadku nie trzeba zapamiętywać ramek na dysku (dlaczego ?)
  - jeśli błąd strony się często zdarza to działania systemu będzie spowolnione (zjawisko tzw "szamotania") – wtedy lepiej użyć swappingu !!!
- algorytm przydziału początkowego ramek
- algorytmy zastępowania stron (= algorytmy wyboru "ramki ofiary")

...licznik błędów stron w WinXP

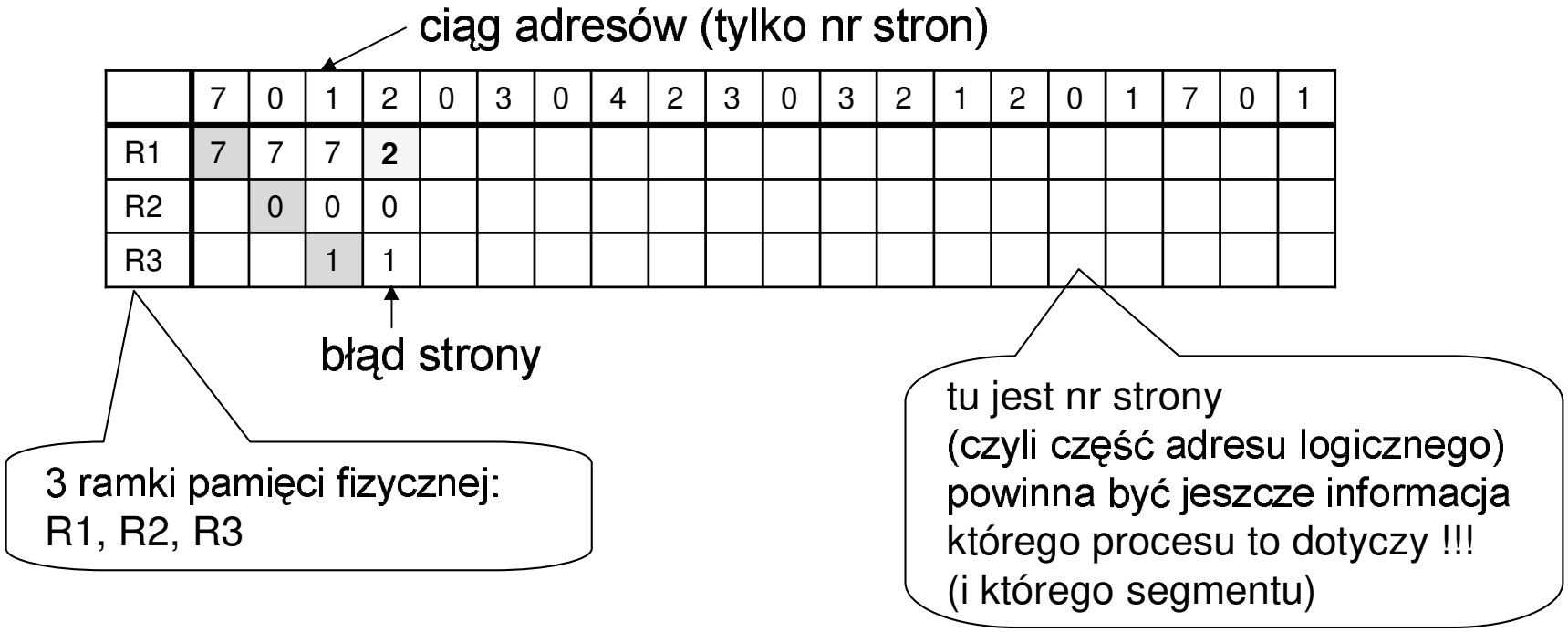
Zarządzanie pamięcią operacyjną;
pamięć wirtualna

- algorytmy zastępowania stron (= algorytmy wyboru "ramki ofiary")
  - **FIFO** [ang. First In First Out]
  - **OPT** – optymalny, ale wymaga "wiedzy o przyszłości"
  - **LRU** [ang. Least Recently Used] – przybliżenie OPT

...licznik błędów stron w WinXP

Zarządzanie pamięcią operacyjną; **pamięć wirtualna**

- algorytmy zastępowania stron:
  - **FIFO [ang. First In First Out]**
    ramka ofiara= ramka która najdawniej została przydzielona stronie;



	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
R1	7	7	7	**2**																
R2		0	0	0																
R3			1	1																

Zarządzanie pamięcią operacyjną;
pamięć wirtualna

- algorytmy zastępowania stron:
  - **FIFO [ang. First In First Out]**
    ramka ofiara= ramka która najdawniej została przydzielona stronie;

	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
R1	7	7	7	2	2															
R2		0	0	0	0															
R3			1	1	1															

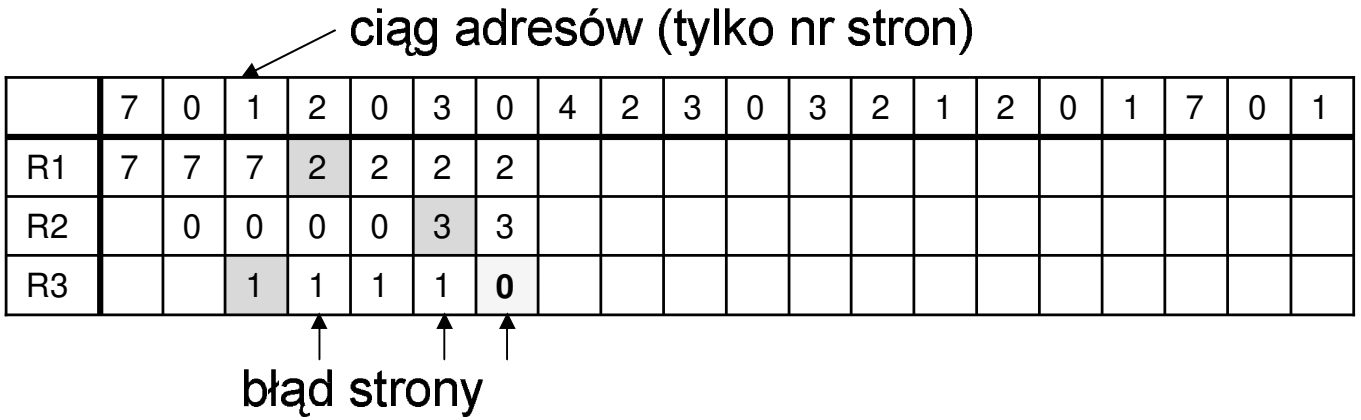
Zarządzanie pamięcią operacyjną; **pamięć wirtualna**

- algorytmy zastępowania stron:
  - **FIFO [ang. First In First Out]**
    ramka ofiara= ramka która najdawniej została przydzielona stronie

	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
R1	7	7	7	2	2	2														
R2		0	0	0	0	**3**														
R3			1	1	1	1														

Zarządzanie pamięcią operacyjną; **pamięć wirtualna**

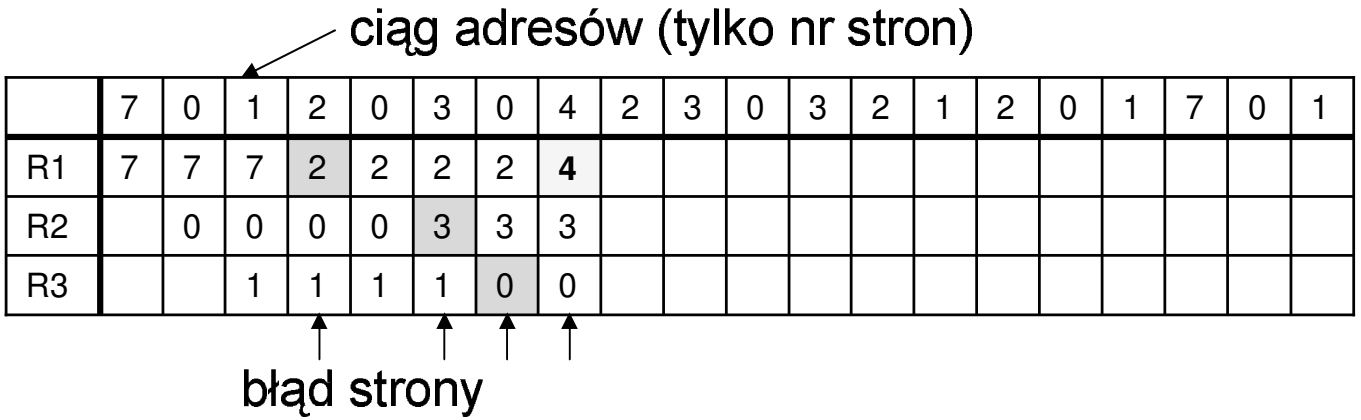
- algorytmy zastępowania stron:
  - **FIFO [ang. First In First Out]**
    ramka ofiara= ramka która najdawniej została przydzielona stronie;



	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
R1	7	7	7	2	2	2	2													
R2		0	0	0	0	3	3													
R3			1	1	1	1	**0**													

Zarządzanie pamięcią operacyjną;
pamięć wirtualna

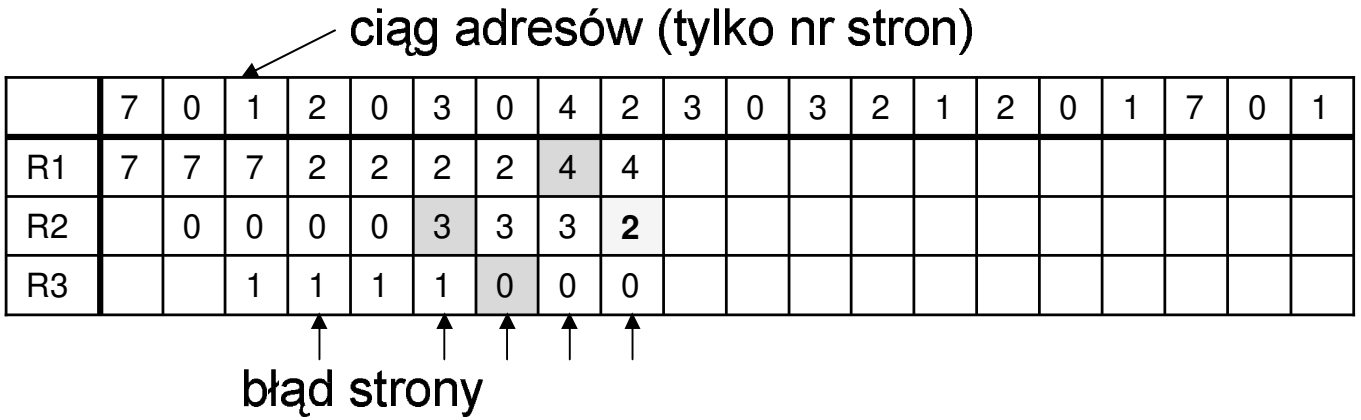
- algorytmy zastępowania stron:
  - **FIFO [ang. First In First Out]**
    ramka ofiara= ramka która najdawniej została przydzielona stronie;



	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
R1	7	7	7	2	2	2	2	**4**												
R2		0	0	0	0	3	3	3												
R3			1	1	1	1	0	0												

Zarządzanie pamięcią operacyjną; **pamięć wirtualna**

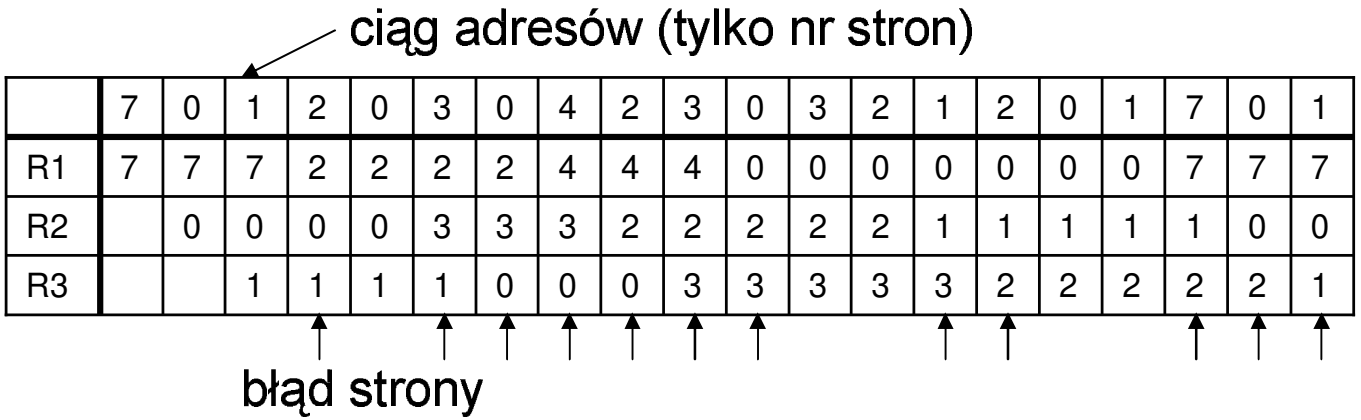
- algorytmy zastępowania stron:
  - **FIFO [ang. First In First Out]**
    ramka ofiara= ramka która najdawniej została przydzielona stronie;



	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
R1	7	7	7	2	2	2	2	4	4											
R2		0	0	0	0	3	3	3	**2**											
R3			1	1	1	1	0	0	0											

Zarządzanie pamięcią operacyjną;
pamięć wirtualna

- algorytmy zastępowania stron:
  - **FIFO [ang. First In First Out]**
    ramka ofiara= ramka która najdawniej została przydzielona stronie;

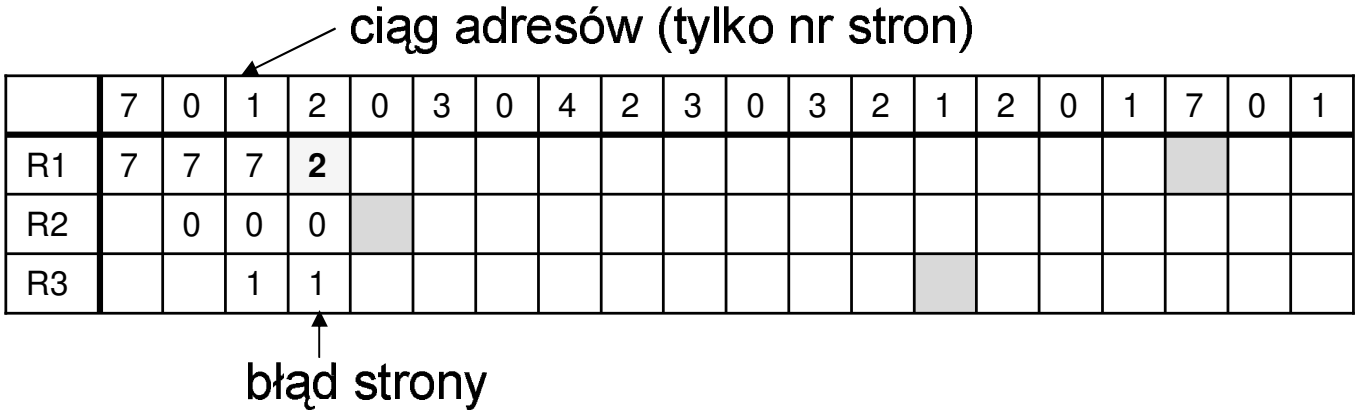


	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
R1	7	7	7	2	2	2	2	4	4	4	0	0	0	0	0	0	0	7	7	7
R2		0	0	0	0	3	3	3	2	2	2	2	2	1	1	1	1	1	0	0
R3			1	1	1	1	0	0	0	3	3	3	3	3	2	2	2	2	2	1

(liczba błędów stron w powyższym przykładzie = 12)

Zarządzanie pamięcią operacyjną;
pamięć wirtualna

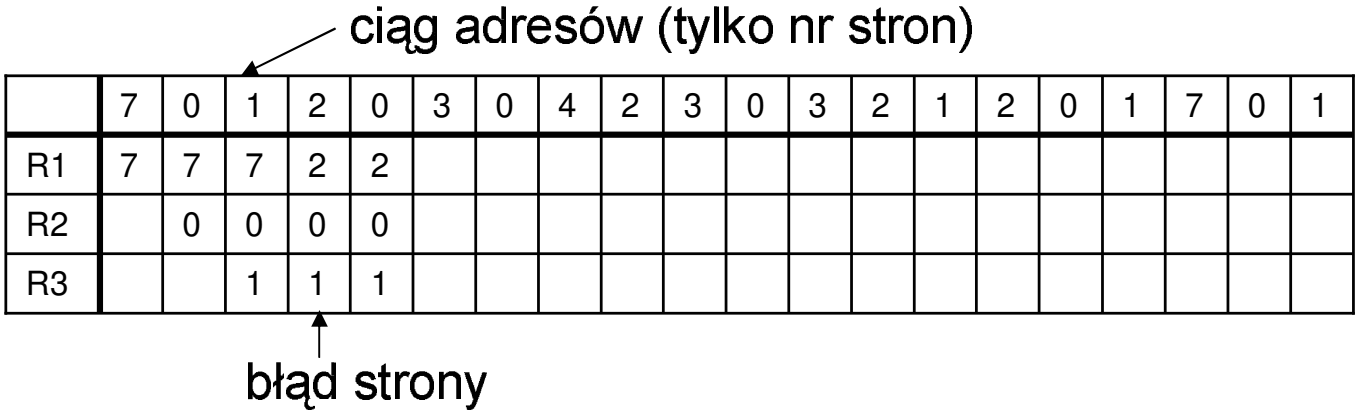
- algorytmy zastępowania stron:
  - **OPT algorytm optymalny (trzeba "znać przyszłości")**
    ramka ofiara= ramkę która najdłużej NIE będzie używana w przyszłości



	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
R1	7	7	7	**2**																
R2		0	0	0																
R3			1	1																

Zarządzanie pamięcią operacyjną;
pamięć wirtualna

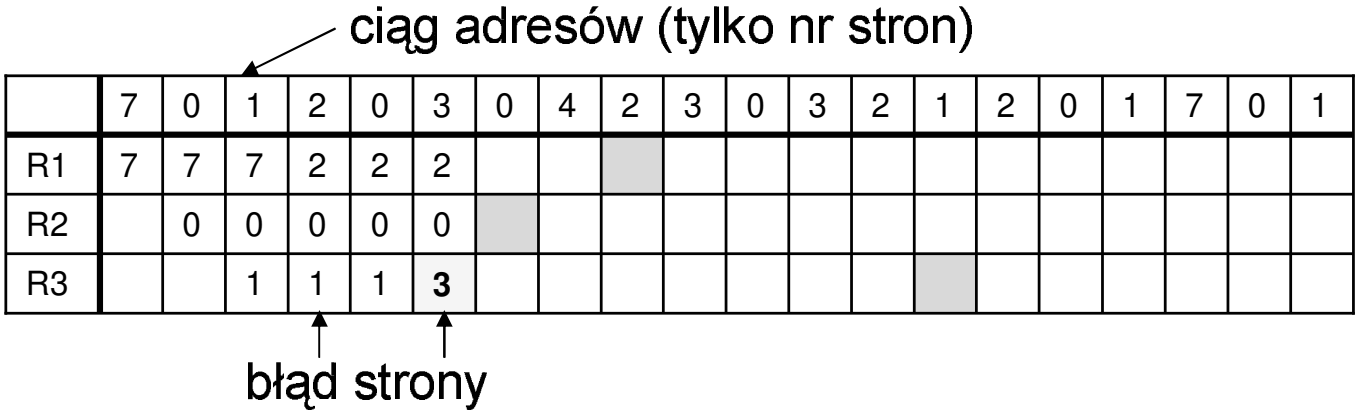
- algorytmy zastępowania stron:
  - **OPT algorytm optymalny (trzeba "znać przyszłości")**
    ramka ofiara= ramkę która najdłużej NIE będzie używana w przyszłości



	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
R1	7	7	7	2	2															
R2		0	0	0	0															
R3			1	1	1															

Zarządzanie pamięcią operacyjną;
pamięć wirtualna

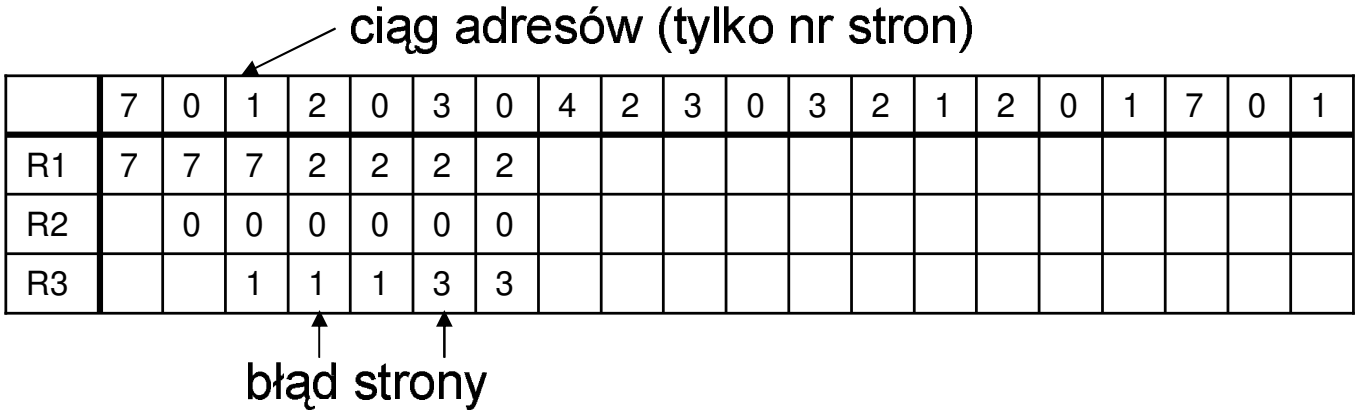
- algorytmy zastępowania stron:
  - **OPT algorytm optymalny (trzeba "znać przyszłości")**
    ramka ofiara= ramkę która najdłużej NIE będzie używana w przyszłości



	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
R1	7	7	7	2	2	2														
R2		0	0	0	0	0														
R3			1	1	1	**3**														

Zarządzanie pamięcią operacyjną;
pamięć wirtualna

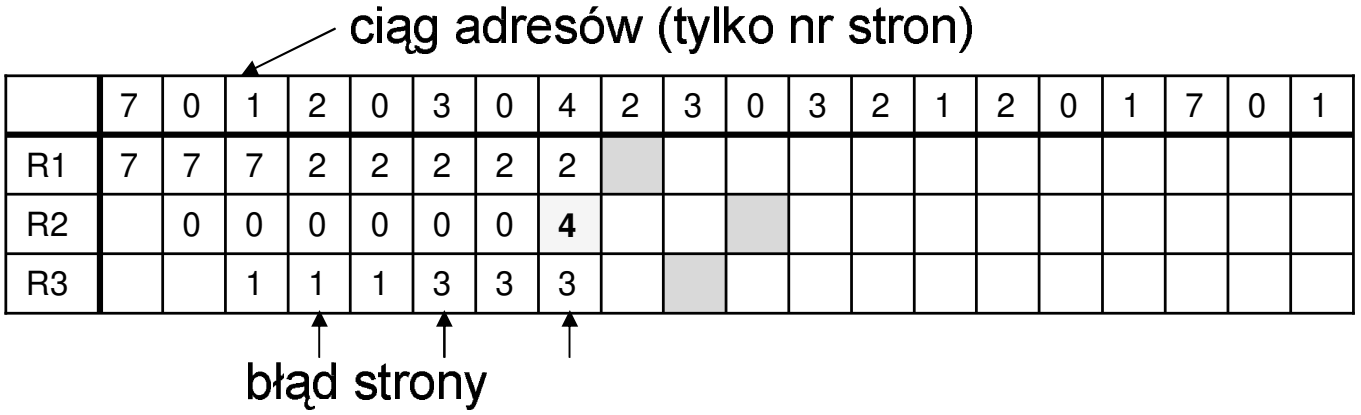
- algorytmy zastępowania stron:
  - **OPT algorytm optymalny (trzeba "znać przyszłości")**
    ramka ofiara= ramkę która najdłużej NIE będzie używana w przyszłości



	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
R1	7	7	7	2	2	2	2													
R2		0	0	0	0	0	0													
R3			1	1	1	3	3													

Zarządzanie pamięcią operacyjną;
pamięć wirtualna

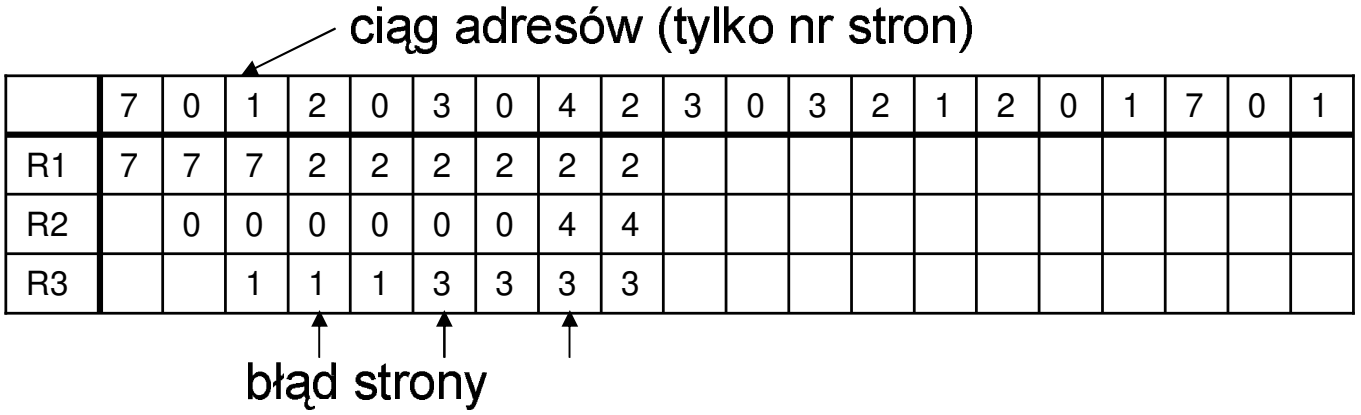
- algorytmy zastępowania stron:
  - **OPT algorytm optymalny (trzeba "znać przyszłości")**
    ramka ofiara= ramkę która najdłużej NIE będzie używana w przyszłości



	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
R1	7	7	7	2	2	2	2	2												
R2		0	0	0	0	0	0	**4**												
R3			1	1	1	3	3	3												

Zarządzanie pamięcią operacyjną;
pamięć wirtualna

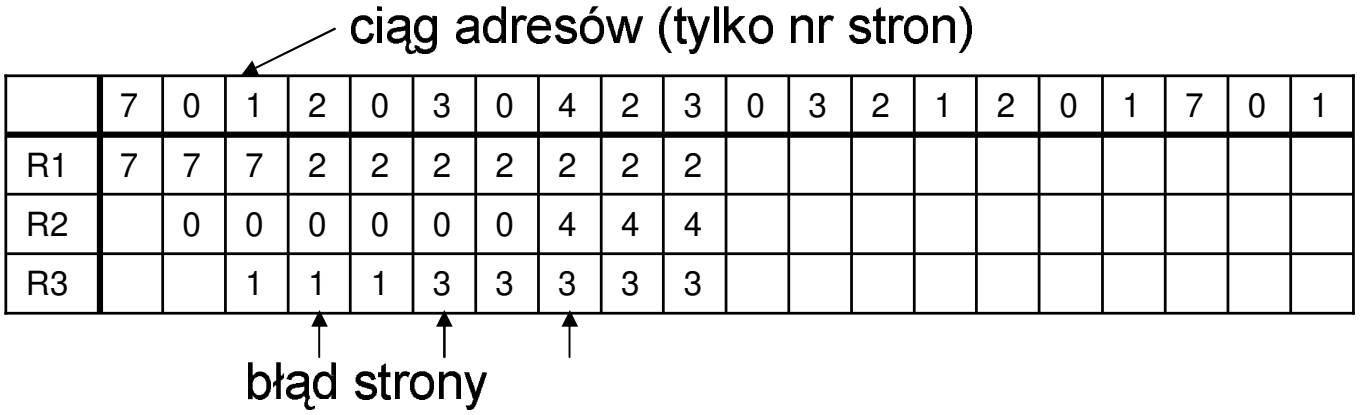
- algorytmy zastępowania stron:
  - **OPT algorytm optymalny (trzeba "znać przyszłości")**
    ramka ofiara= ramkę która najdłużej NIE będzie używana w przyszłości



	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
R1	7	7	7	2	2	2	2	2	2											
R2		0	0	0	0	0	0	4	4											
R3			1	1	1	3	3	3	3											

Zarządzanie pamięcią operacyjną;
pamięć wirtualna

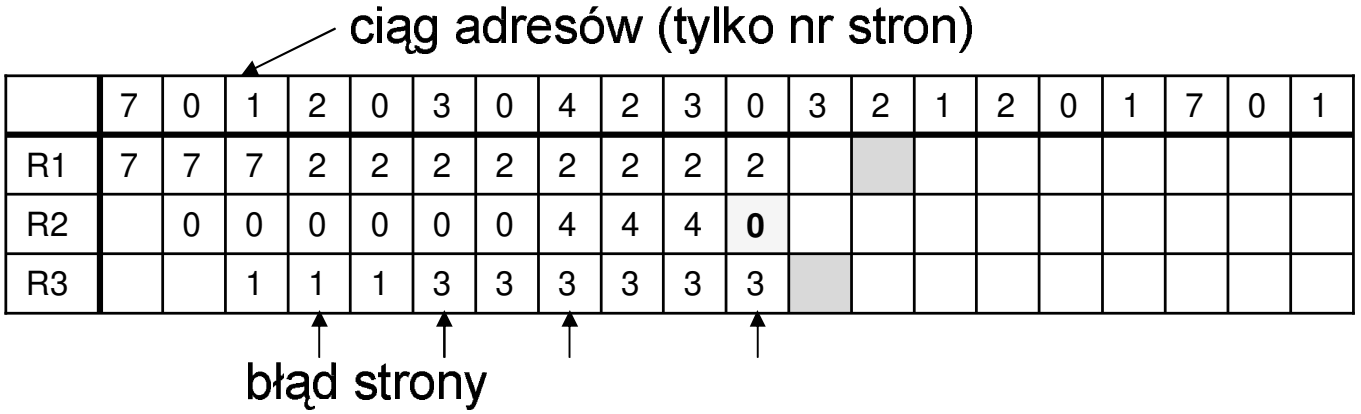
- algorytmy zastępowania stron:
  - **OPT algorytm optymalny (trzeba "znać przyszłości")**
    ramka ofiara= ramkę która najdłużej NIE będzie używana w przyszłości



	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
R1	7	7	7	2	2	2	2	2	2	2										
R2		0	0	0	0	0	0	4	4	4										
R3			1	1	1	3	3	3	3	3										

Zarządzanie pamięcią operacyjną;
pamięć wirtualna

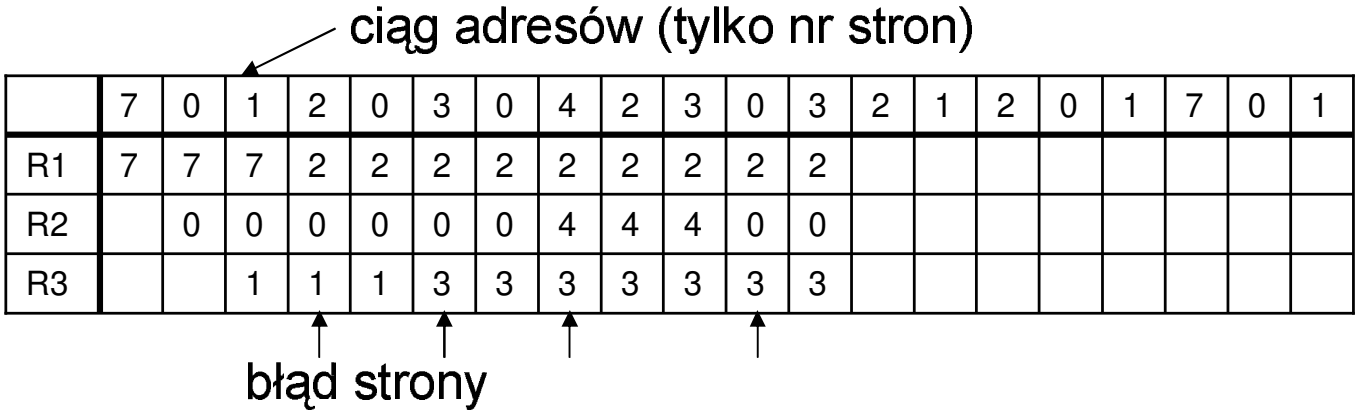
- algorytmy zastępowania stron:
  - **OPT algorytm optymalny (trzeba "znać przyszłości")**
    ramka ofiara= ramkę która najdłużej NIE będzie używana w przyszłości



	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
R1	7	7	7	2	2	2	2	2	2	2	2									
R2		0	0	0	0	0	0	4	4	4	**0**									
R3			1	1	1	3	3	3	3	3	3									

Zarządzanie pamięcią operacyjną;
pamięć wirtualna

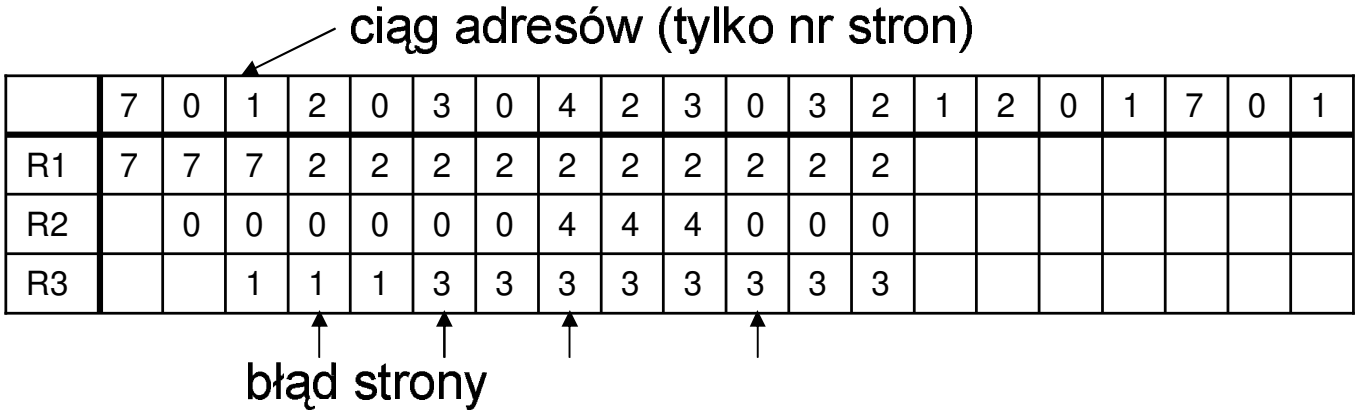
- algorytmy zastępowania stron:
  - **OPT algorytm optymalny (trzeba "znać przyszłości")**
    ramka ofiara= ramkę która najdłużej NIE będzie używana w przyszłości



	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
R1	7	7	7	2	2	2	2	2	2	2	2	2								
R2		0	0	0	0	0	0	4	4	4	0	0								
R3			1	1	1	3	3	3	3	3	3	3								

Zarządzanie pamięcią operacyjną;
pamięć wirtualna

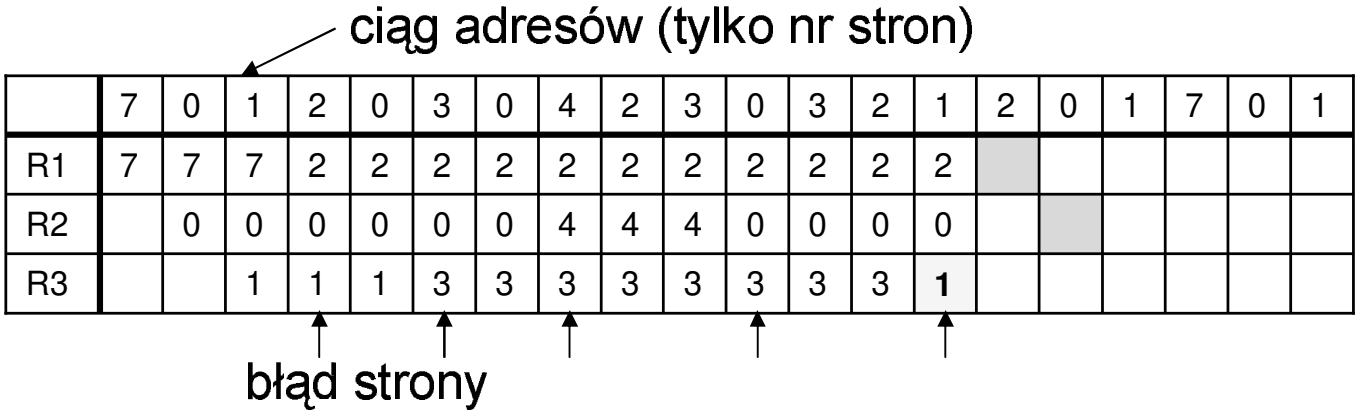
- algorytmy zastępowania stron:
  - **OPT algorytm optymalny (trzeba "znać przyszłości")**
    ramka ofiara= ramkę która najdłużej NIE będzie używana w przyszłości



	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
R1	7	7	7	2	2	2	2	2	2	2	2	2	2							
R2		0	0	0	0	0	0	4	4	4	0	0	0							
R3			1	1	1	3	3	3	3	3	3	3	3							

Zarządzanie pamięcią operacyjną;
pamięć wirtualna

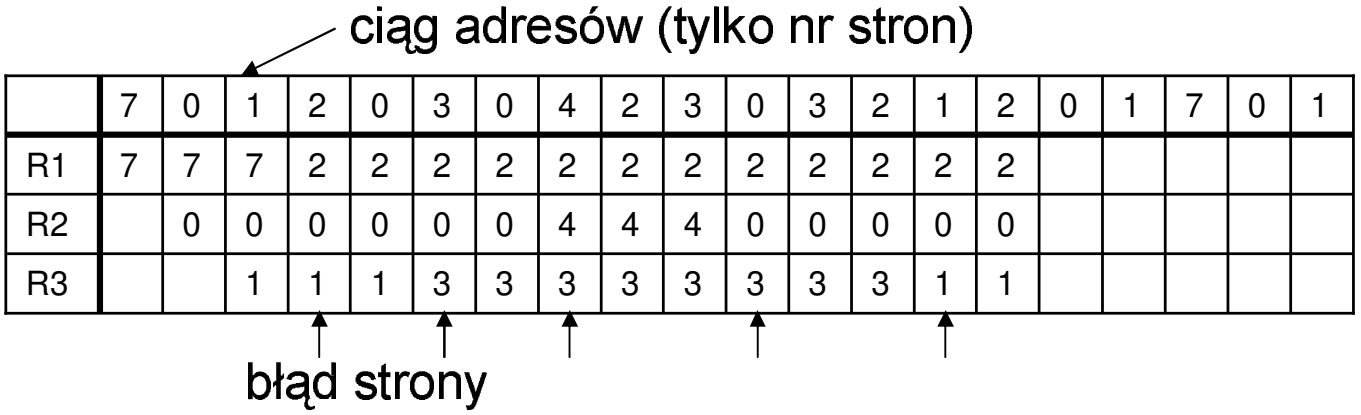
- algorytmy zastępowania stron:
  - **OPT algorytm optymalny (trzeba "znać przyszłości")**
    ramka ofiara= ramkę która najdłużej NIE będzie używana w przyszłości



	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
R1	7	7	7	2	2	2	2	2	2	2	2	2	2	2						
R2		0	0	0	0	0	0	4	4	4	0	0	0	0						
R3			1	1	1	3	3	3	3	3	3	3	3	**1**						

Zarządzanie pamięcią operacyjną;
pamięć wirtualna

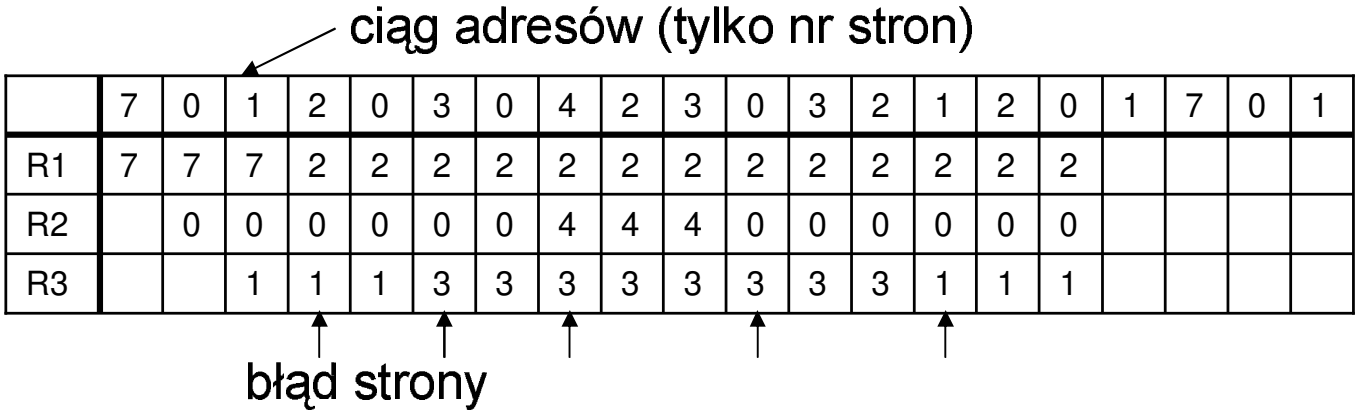
- algorytmy zastępowania stron:
  - **OPT algorytm optymalny (trzeba "znać przyszłości")**
    ramka ofiara= ramkę która najdłużej NIE będzie używana w przyszłości



	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
R1	7	7	7	2	2	2	2	2	2	2	2	2	2	2	2					
R2		0	0	0	0	0	0	4	4	4	0	0	0	0	0					
R3			1	1	1	3	3	3	3	3	3	3	3	1	1					

Zarządzanie pamięcią operacyjną;
pamięć wirtualna

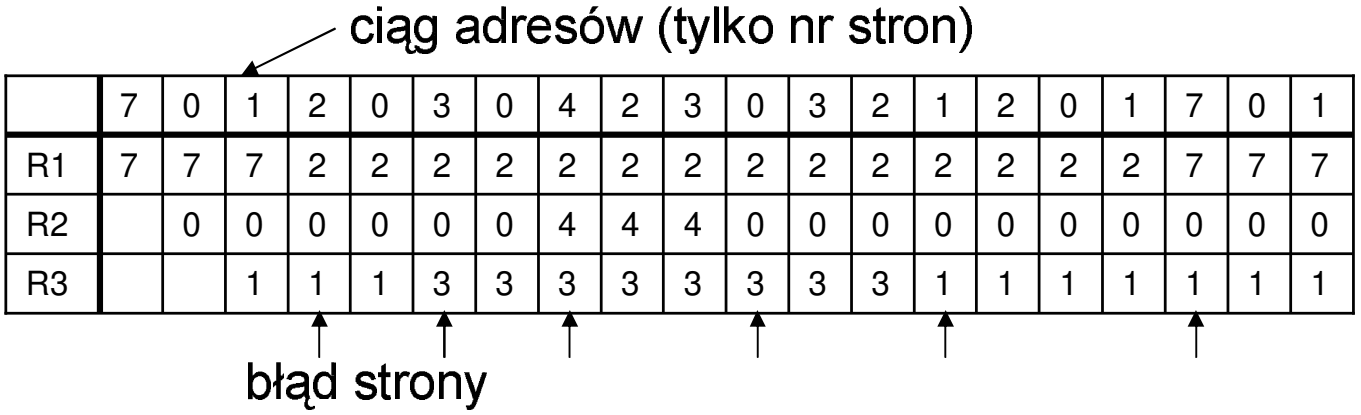
- algorytmy zastępowania stron:
  - **OPT algorytm optymalny (trzeba "znać przyszłości")**
    ramka ofiara= ramkę która najdłużej NIE będzie używana w przyszłości



	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
R1	7	7	7	2	2	2	2	2	2	2	2	2	2	2	2	2				
R2		0	0	0	0	0	0	4	4	4	0	0	0	0	0	0				
R3			1	1	1	3	3	3	3	3	3	3	3	1	1	1				

Zarządzanie pamięcią operacyjną; **pamięć wirtualna**

- algorytmy zastępowania stron:
  - **OPT algorytm optymalny (trzeba "znać przyszłości")**
    ramka ofiara= ramkę która najdłużej NIE będzie używana w przyszłości

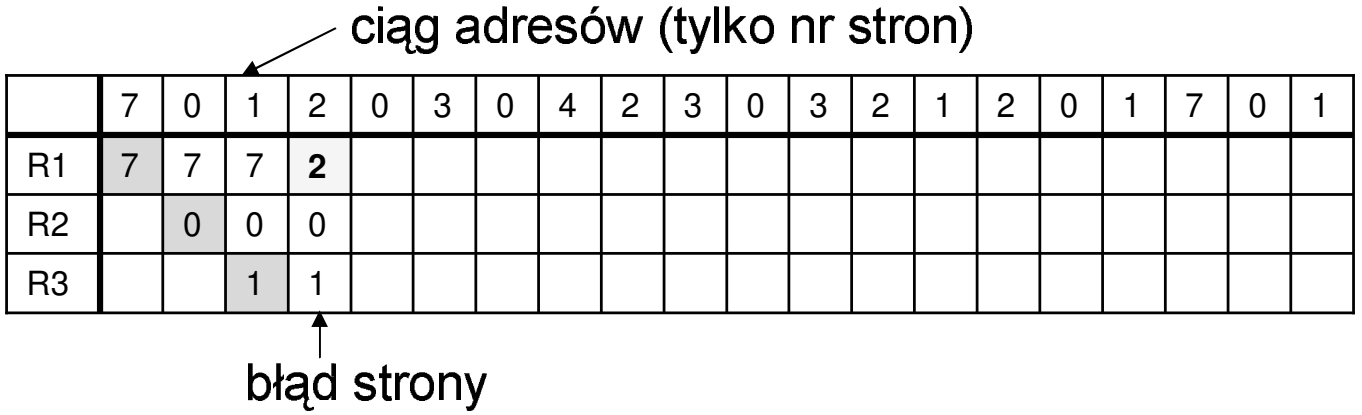


	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
R1	7	7	7	2	2	2	2	2	2	2	2	2	2	2	2	2	2	7	7	7
R2		0	0	0	0	0	0	4	4	4	0	0	0	0	0	0	0	0	0	0
R3			1	1	1	3	3	3	3	3	3	3	3	1	1	1	1	1	1	1

(liczba błędów stron w powyższym przykładzie = 6)

Zarządzanie pamięcią operacyjną;
pamięć wirtualna

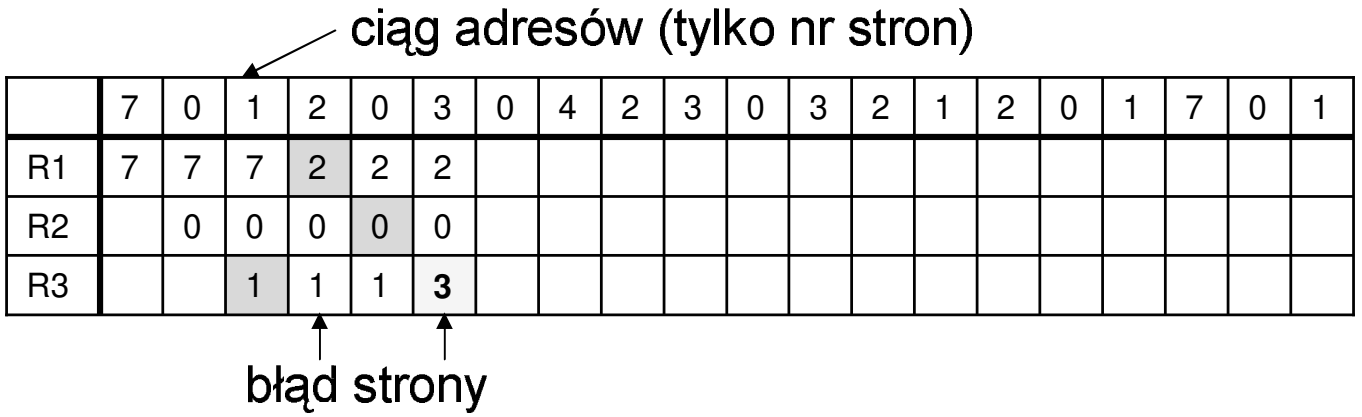
- algorytmy zastępowania stron:
  - **LRU [ang. Least Recently Used]**
    ramka ofiara= ramka która najdawniej była używana (był dostęp do strony)



	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
R1	7	7	7	**2**																
R2		0	0	0																
R3			1	1																

Zarządzanie pamięcią operacyjną;
pamięć wirtualna

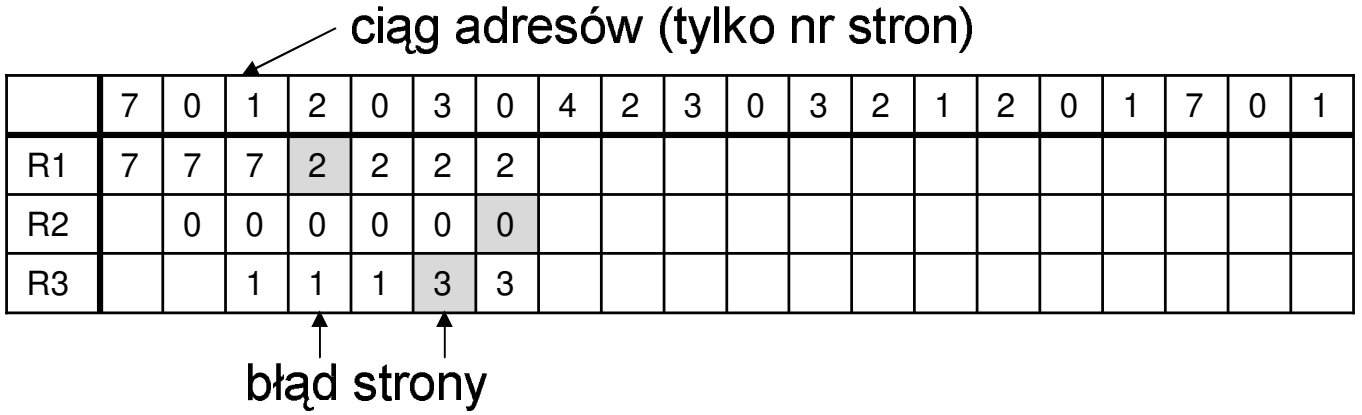
- algorytmy zastępowania stron:
  - **LRU [ang. Least Recently Used]**
    ramka ofiara= ramka która najdawniej była używana (był dostęp do strony)



	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
R1	7	7	7	2	2	2														
R2		0	0	0	0	0														
R3			1	1	1	**3**														

Zarządzanie pamięcią operacyjną;
pamięć wirtualna

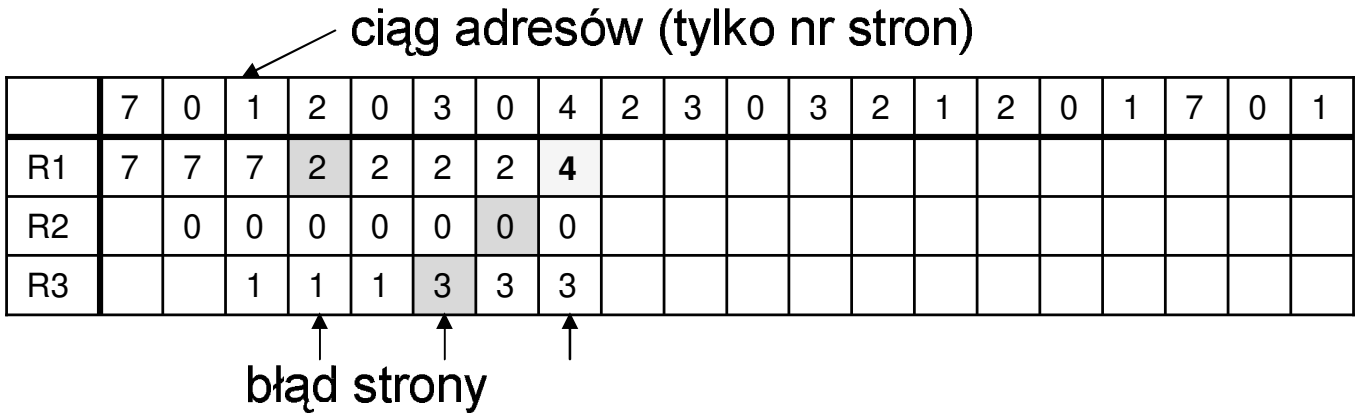
- algorytmy zastępowania stron:
  - **LRU [ang. Least Recently Used]**
    ramka ofiara= ramka która najdawniej była używana (był dostęp do strony)



	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
R1	7	7	7	2	2	2	2													
R2		0	0	0	0	0	0													
R3			1	1	1	3	3													

Zarządzanie pamięcią operacyjną;
pamięć wirtualna

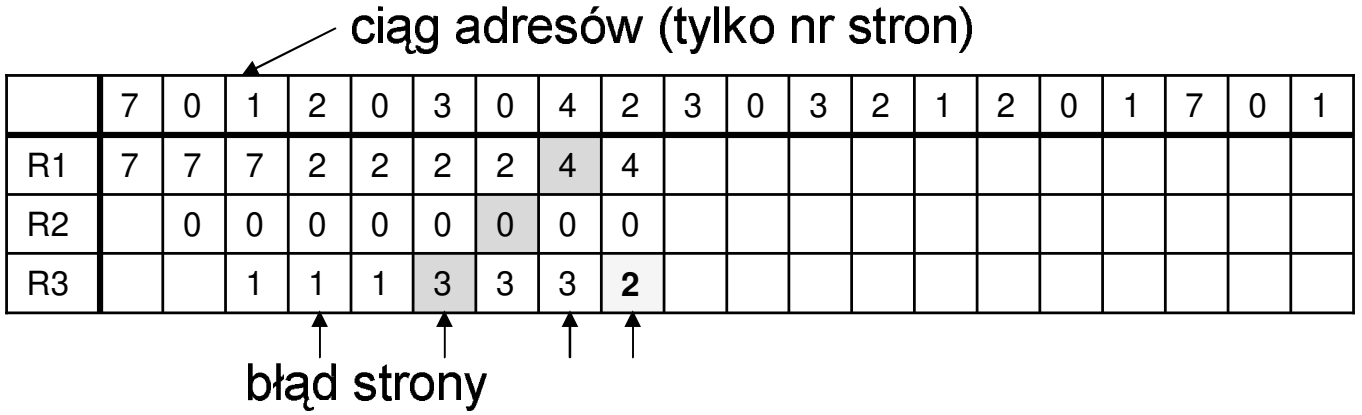
- algorytmy zastępowania stron:
  - **LRU [ang. Least Recently Used]**
    ramka ofiara= ramka która najdawniej była używana (był dostęp do strony)



	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
R1	7	7	7	2	2	2	2	**4**												
R2		0	0	0	0	0	0	0												
R3			1	1	1	3	3	3												

Zarządzanie pamięcią operacyjną;
pamięć wirtualna

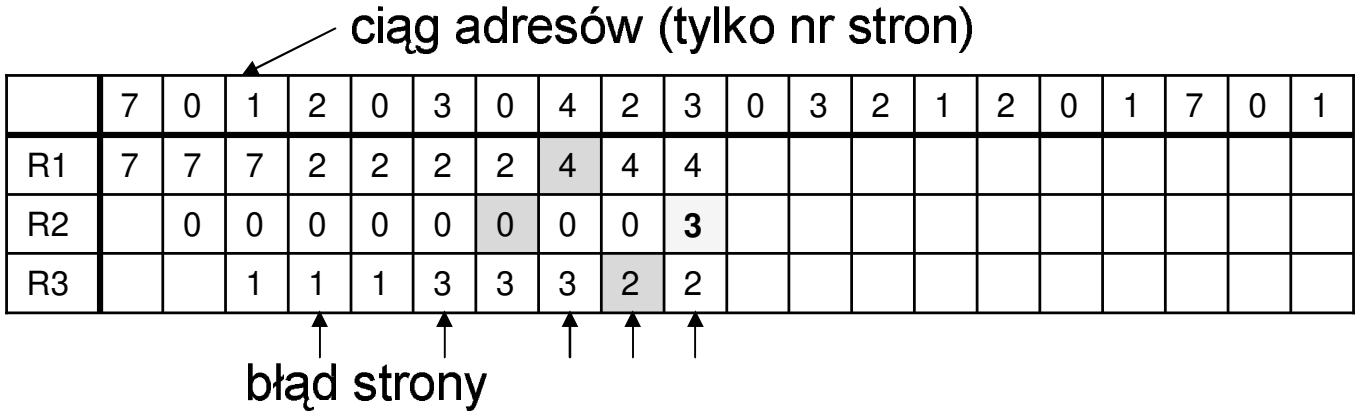
- algorytmy zastępowania stron:
  - **LRU [ang. Least Recently Used]**
    ramka ofiara= ramka która najdawniej była używana (był dostęp do strony)



	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
R1	7	7	7	2	2	2	2	4	4											
R2		0	0	0	0	0	0	0	0											
R3			1	1	1	3	3	3	**2**											

Zarządzanie pamięcią operacyjną;
pamięć wirtualna

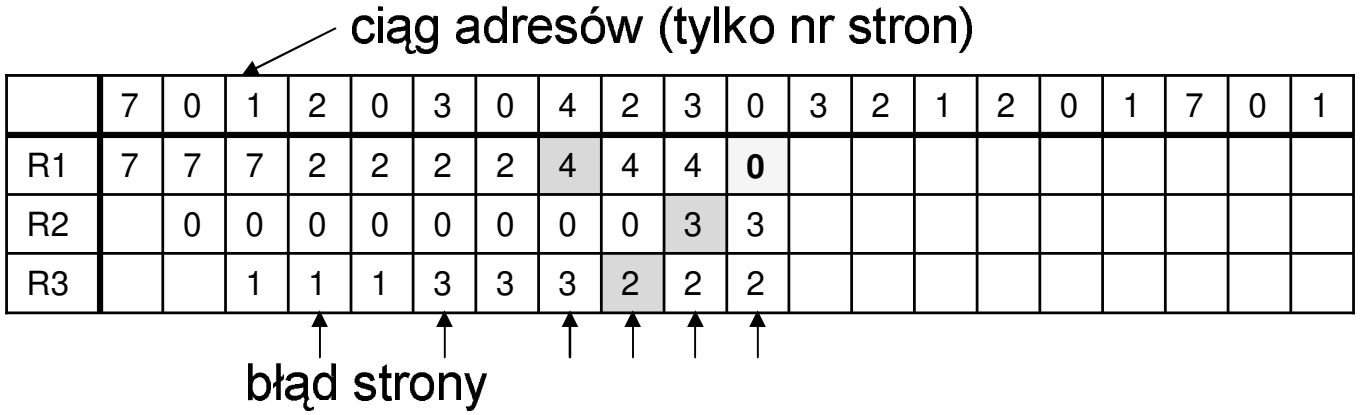
- algorytmy zastępowania stron:
  - **LRU [ang. Least Recently Used]**
    ramka ofiara= ramka która najdawniej była używana (był dostęp do strony)



	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
R1	7	7	7	2	2	2	2	4	4	4										
R2		0	0	0	0	0	0	0	0	**3**										
R3			1	1	1	3	3	3	2	2										

Zarządzanie pamięcią operacyjną;
pamięć wirtualna

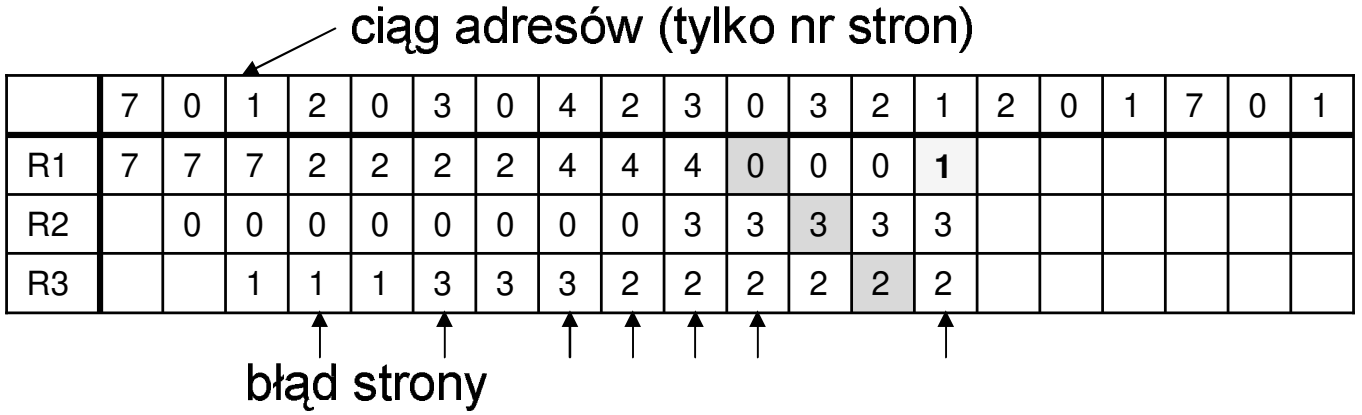
- algorytmy zastępowania stron:
  - **LRU [ang. Least Recently Used]**
    ramka ofiara= ramka która najdawniej była używana (był dostęp do strony)



	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
R1	7	7	7	2	2	2	2	4	4	4	**0**									
R2		0	0	0	0	0	0	0	0	3	3									
R3			1	1	1	3	3	3	2	2	2									

Zarządzanie pamięcią operacyjną;
pamięć wirtualna

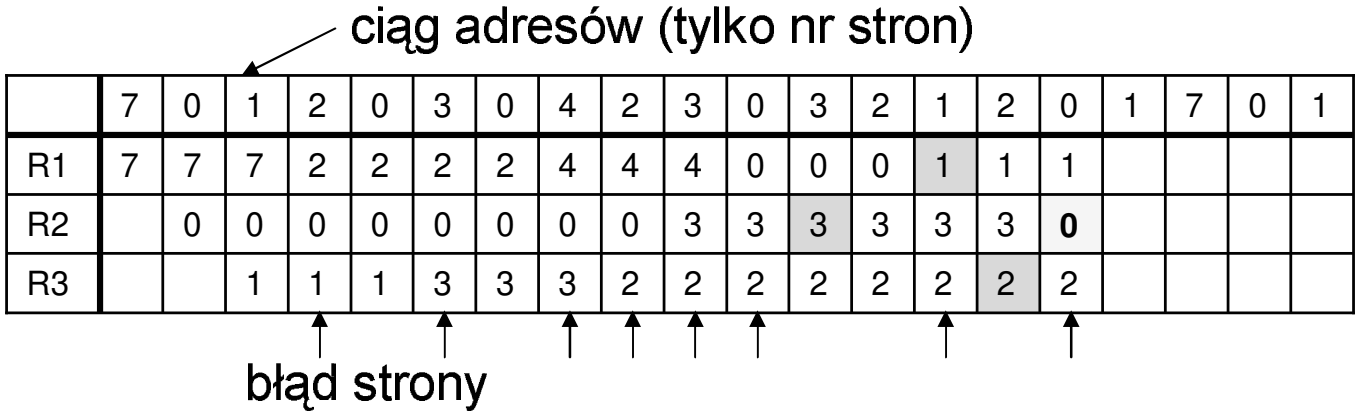
- algorytmy zastępowania stron:
  - **LRU [ang. Least Recently Used]**
    ramka ofiara= ramka która najdawniej była używana (był dostęp do strony)



	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
R1	7	7	7	2	2	2	2	4	4	4	0	0	0	**1**						
R2		0	0	0	0	0	0	0	0	3	3	3	3	3						
R3			1	1	1	3	3	3	2	2	2	2	2	2						

Zarządzanie pamięcią operacyjną;
pamięć wirtualna

- algorytmy zastępowania stron:
  - **LRU [ang. Least Recently Used]**
    ramka ofiara= ramka która najdawniej była używana (był dostęp do strony)



	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
R1	7	7	7	2	2	2	2	4	4	4	0	0	0	1	1	1				
R2		0	0	0	0	0	0	0	0	3	3	3	3	3	3	**0**				
R3			1	1	1	3	3	3	2	2	2	2	2	2	2	2				

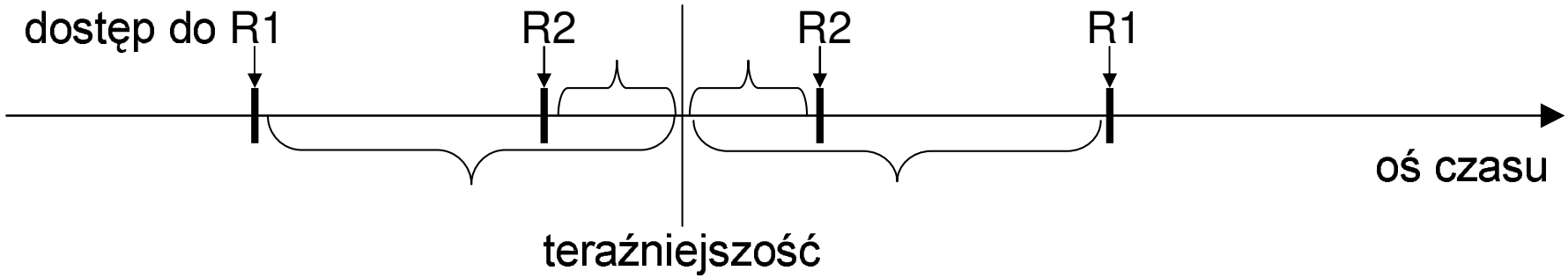
Zarządzanie pamięcią operacyjną; **pamięć wirtualna**

- algorytmy zastępowania stron:
  - **LRU [ang. Least Recently Used]**
    ramka ofiara= ramka która najdawniej była używana (był dostęp do strony)

ciąg adresów (tylko nr stron)

	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
R1	7	7	7	2	2	2	2	4	4	4	0	0	0	1	1	1	1	1	1	1
R2		0	0	0	0	0	0	0	0	3	3	3	3	3	3	0	0	0	0	0
R3			1	1	1	3	3	3	2	2	2	2	2	2	2	2	2	**7**	7	7

błąd strony

(liczba błędów stron = 9)

- dlaczego LRU jest przybliżeniem OPT ???
  - zakładamy że dostęp do ramek ma niezmienną częstotliwość, wtedy ...