

Systemy operacyjne

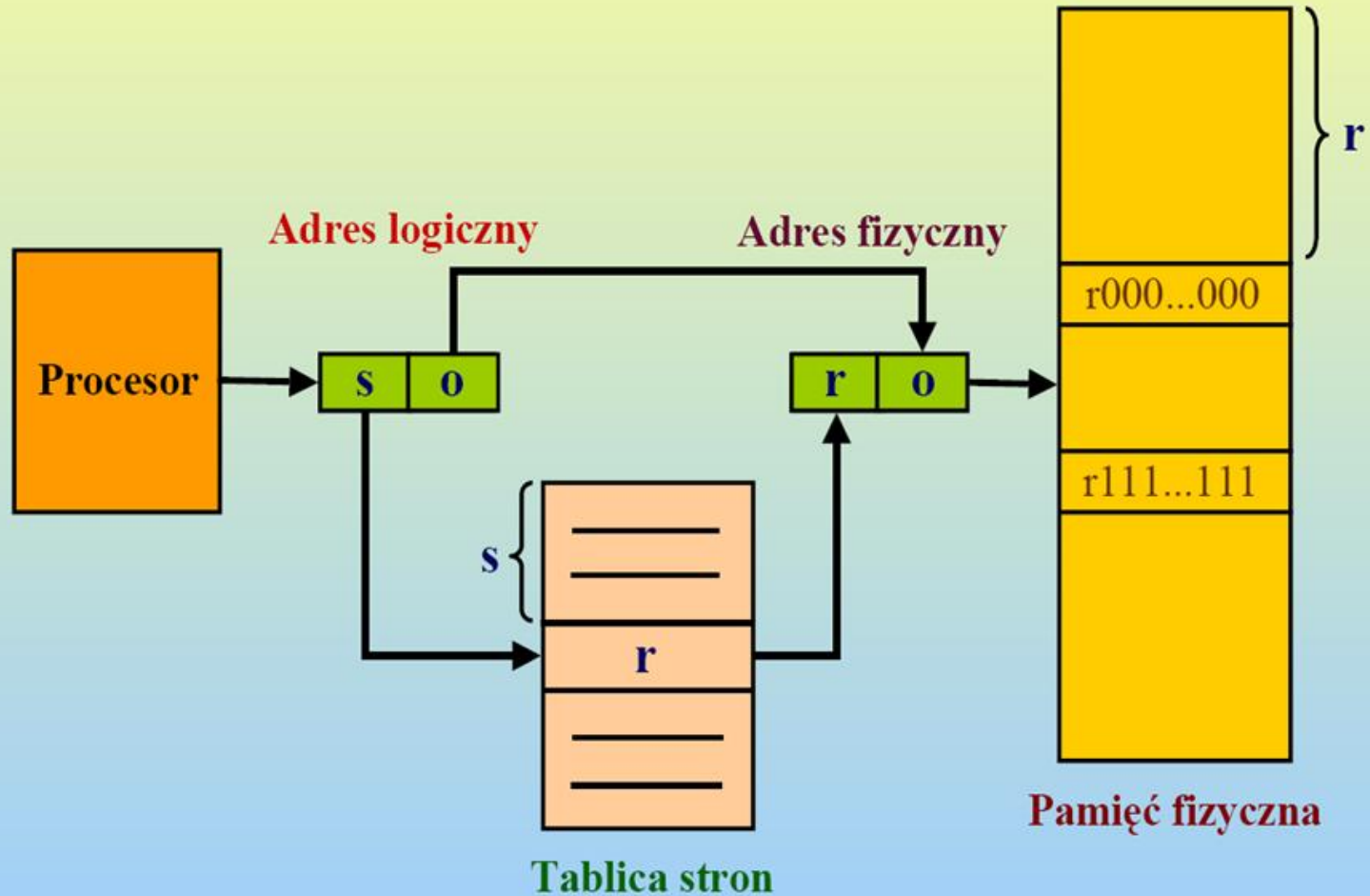
WYKŁAD 12

dr inż. Stanisława Plichta
splichta@ans-ns.edu.pl

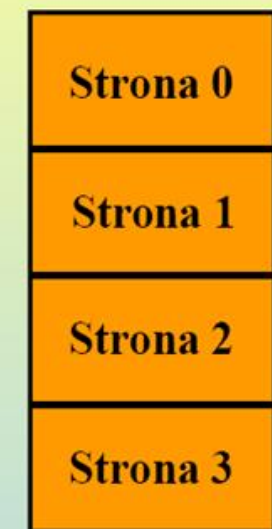
Zarządzanie pamięcią operacyjną

- Adres logiczny - wytworzony przez procesor (adres wirtualny)
- Zbiór wszystkich adresów logicznych - **logiczna przestrzeń adresowa**
- Adres fizyczny - umieszczony w rejestrze adresowym pamięci
- Zbiór wszystkich adresów fizycznych - **fizyczna przestrzeń adresowa**
- Odwzorowanie adresów wirtualnych na fizyczne - **MMU**
(jednostka zarządzająca pamięcią – *memory management unit*)

Odwzorowanie adresu



Przykład stronicowania

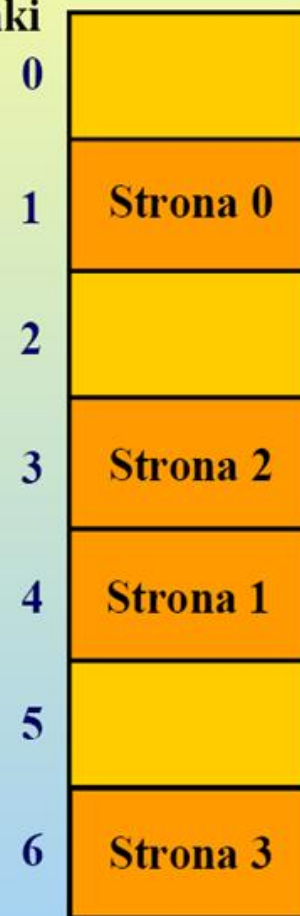


Pamięć logiczna

0	1
1	4
2	3
3	6

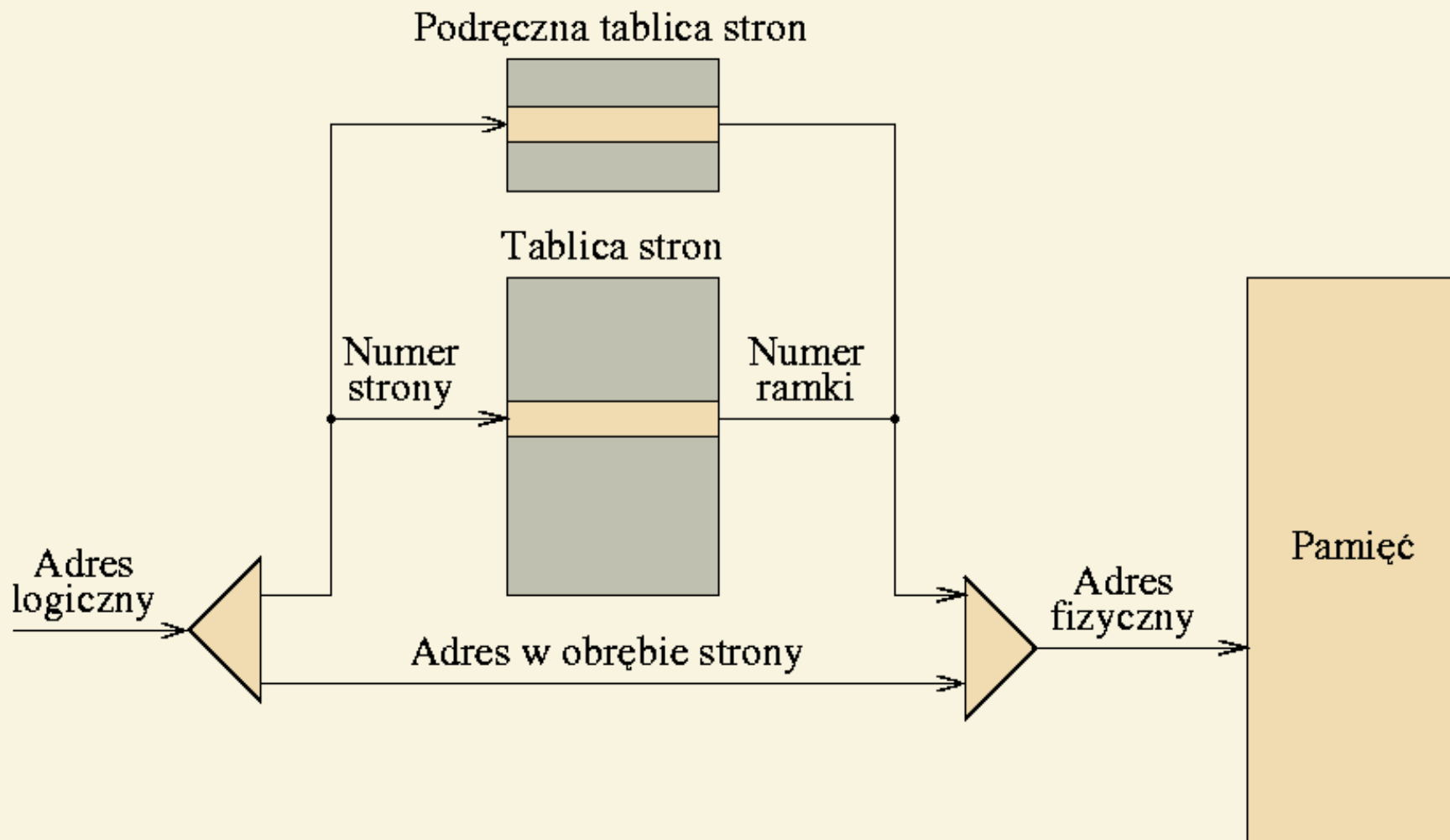
Tablica
stron

Numer
ramki

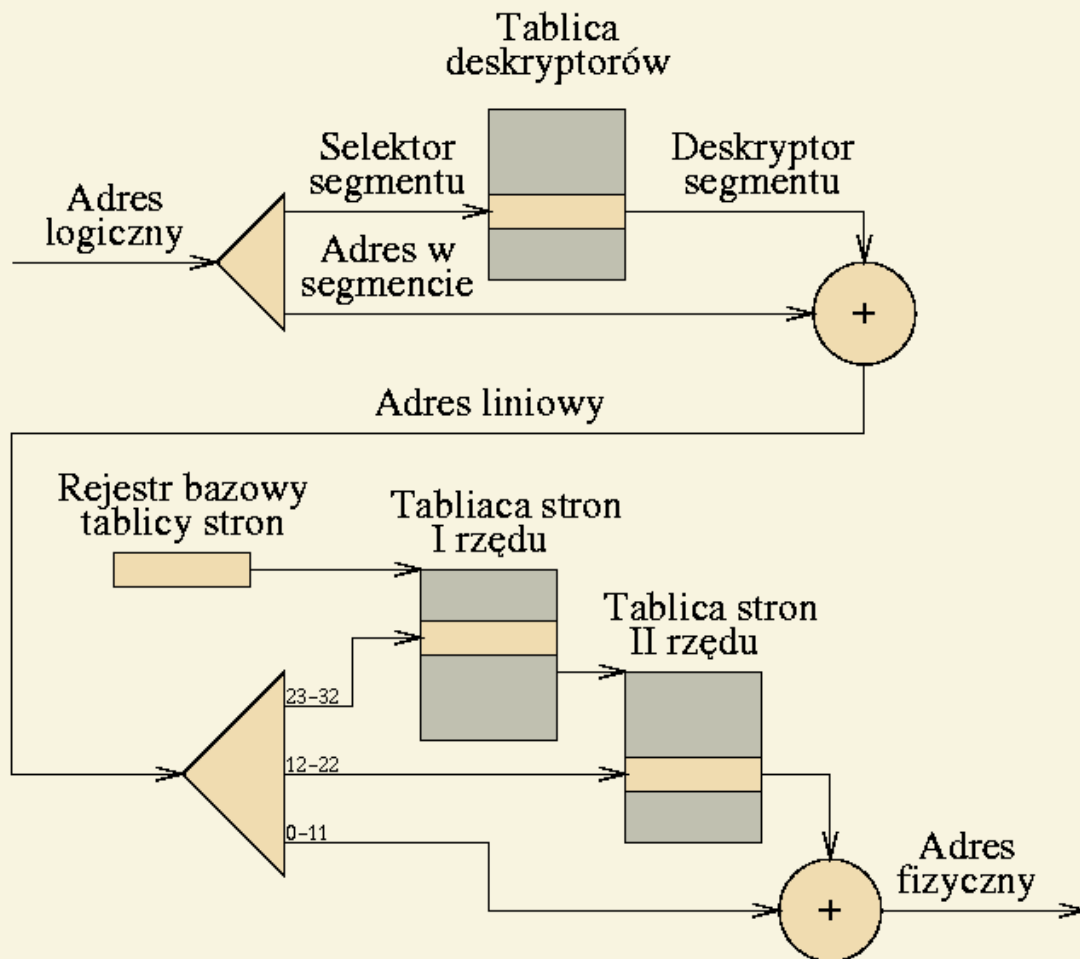


Pamięć fizyczna

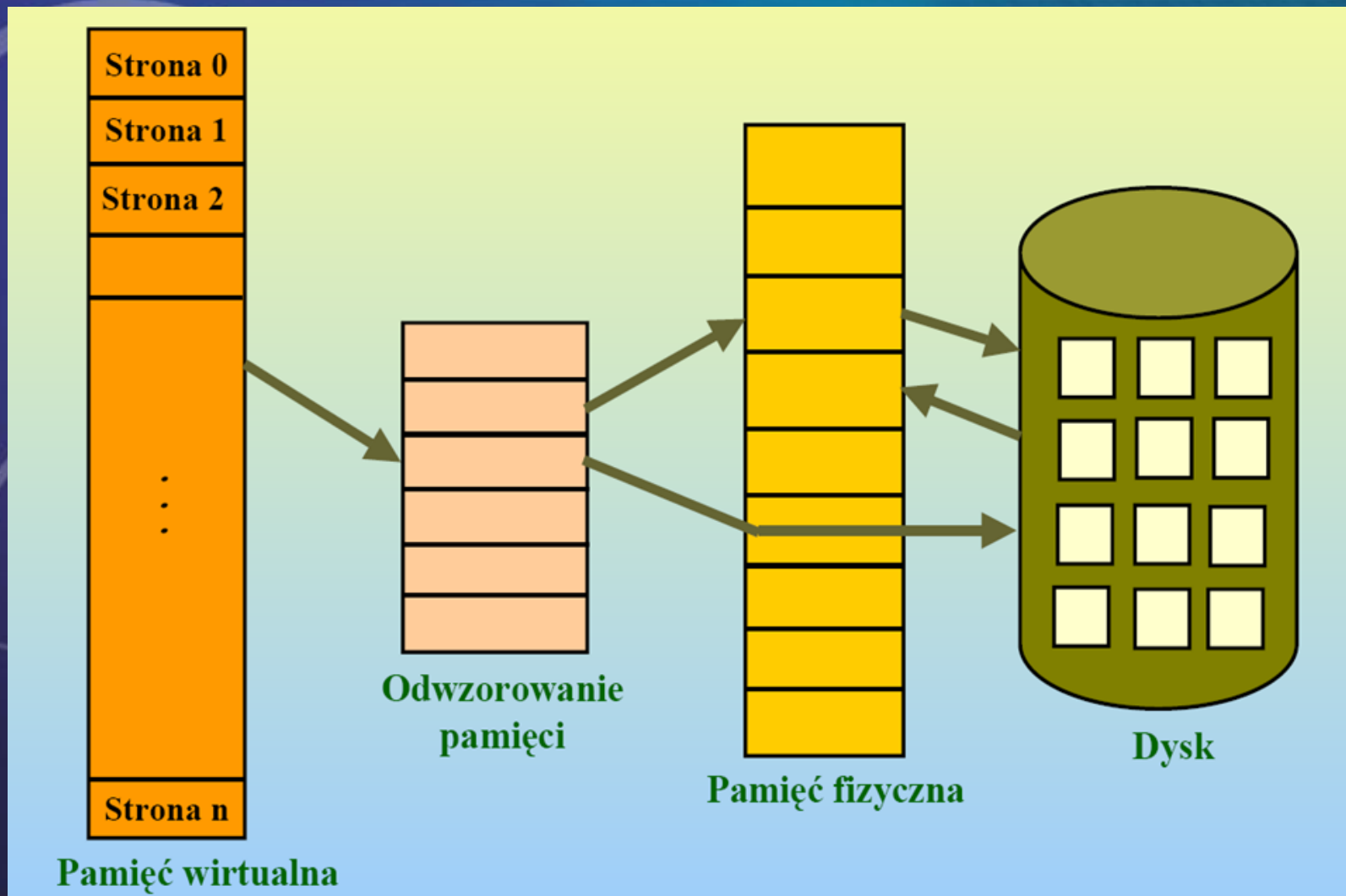
Schemat działania MMU dla stronicowania



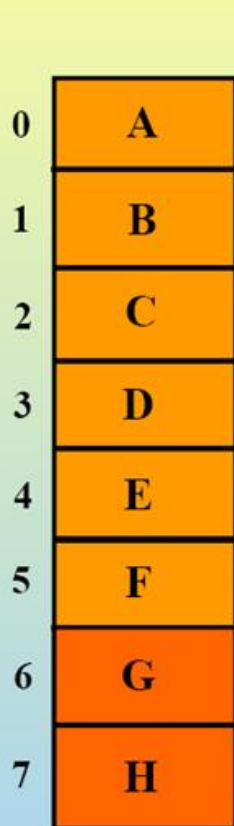
Schemat działania MMU dla stronicowania



Schemat pamięci wirtualnej



Tablica stron z brakami stron w pamięci



Pamięć logiczna

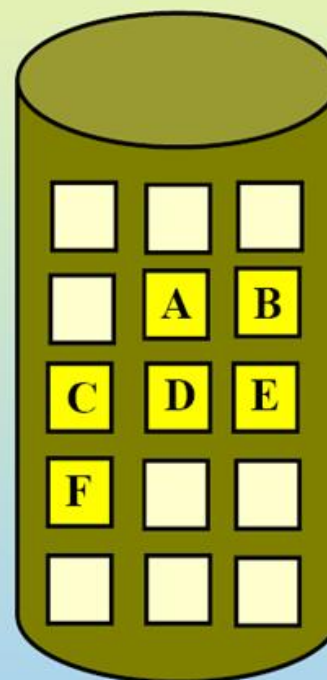
Diagram illustrating the mapping of logical memory pages to physical memory frames (Ramka) and the validity bit (Bit poprawności).

	Ramka	Bit poprawności
0	4	1
1		0
2	6	1
3		0
4		0
5	9	1
6		0
7		0

Tablica stron

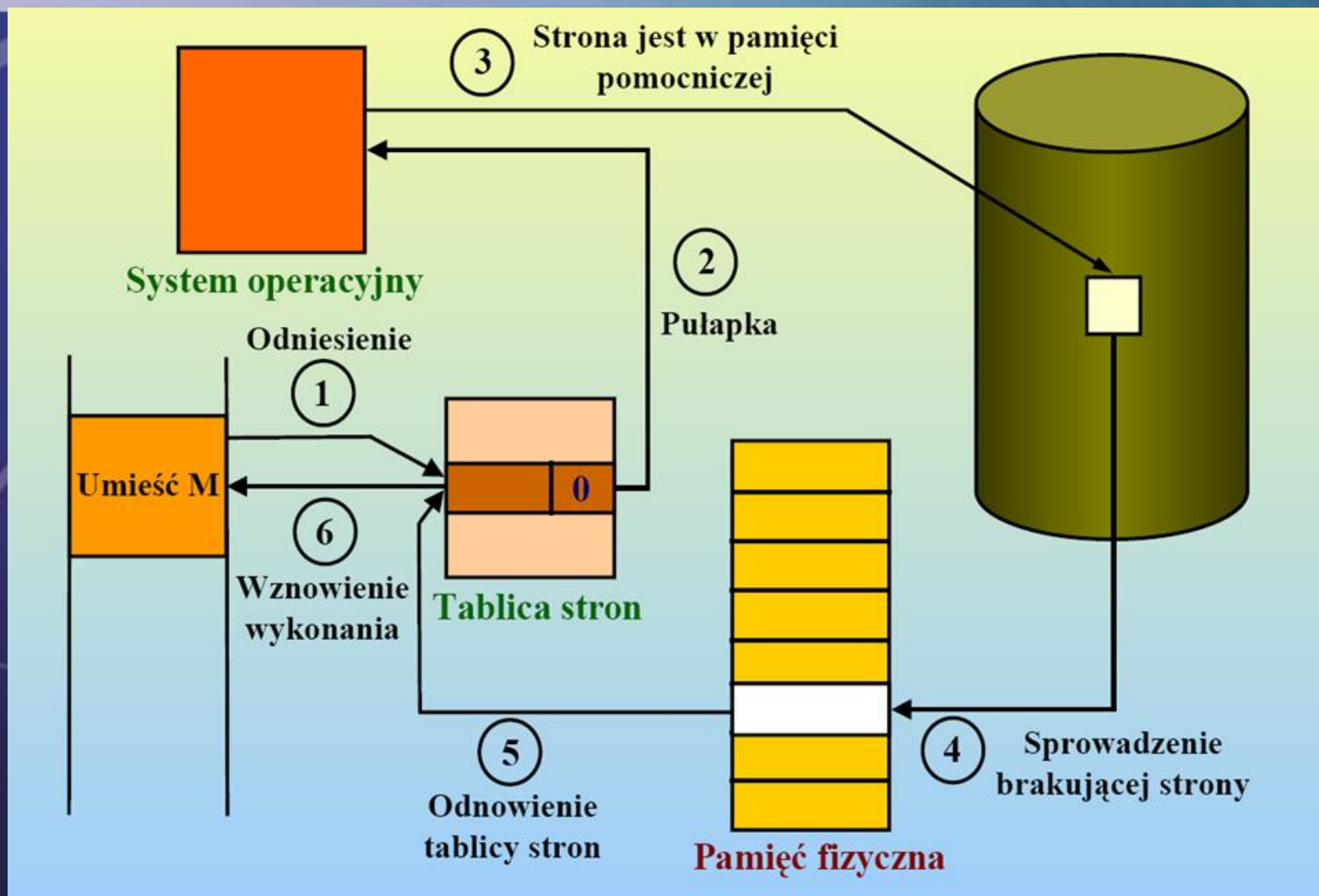


Pamięć fizyczna



Dysk

Etapy obsługi braku strony



Efektywny czas dostępu

Oznaczmy przez p prawdopodobieństwo braku strony

- Efektywny czas dostępu (Effective Access Time (EAT) do pamięci stronicowanej – średni czas wymagany na obsługę odwołania do pamięci (pewna miara sprawności).*

$EAT =$

$(1 - p)$	*	czas dostępu do pamięci
$+ p$	*	(narzut związany z przzerwaniem braku strony + [zapisanie na dysk strony ze zwalnianej ramki] + wczytanie z dysku żądanej strony + narzut związany ze wznowieniem procesu)

Efektywny czas dostępu

Przyjmijmy następujące założenia:

- Czas dostępu do konwencjonalnej pamięci to 1 mikrosekunda
- Połowa z zastępowanych stron wymaga zapisu na dysk
- Czas dyskowego zapisu i odczytu strony to 10 milisekund (10000 mikrosekund)
- Zanedbujemy narzuty związane z obsługą przerwania i wznowieniem procesu (są one niewielkie w porównaniu z gigantycznym kosztem operacji dyskowej)

$$EAT = (1 - p) * 1 + p * (50\% * 10000 + 10000) = 1 + 14999 * p$$

EAT jest wprost proporcjonalny do prawdopodobieństwa wystąpienia przerwania braku strony

Schemat zastępowania stron



Algorytm zastępowania stron

- Algorytm zastępowania stron powinien minimalizować **częstość braków stron** (*page-fault rate*).
- Algorytm ocenia się na podstawie wykonania go na pewnym **ciągu odniesień** (*reference string*) do pamięci i zsumowania braków stron.
 - Ciąg odniesień można tworzyć sztucznie lub na podstawie śledzenia danego systemu.
- W poniższych przykładach ciąg odniesień ma postać:
1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5.



Algorytm FIFO zastępowania stron

liczba ramek wynosi 3

Chwila	1	2	3	4	5	6	7	8	9	10	11	12
Odwołanie	1	2	3	4	1	2	5	1	2	3	4	5
Ramka 1	1	1	1	4	4	4	5	5	5	5	5	5
Ramka 2		2	2	2	1	1	1	1	1	3	3	3
Ramka 3			3	3	3	2	2	2	2	2	4	4

ofiara staje się strona, która najdłużej przebywa w pamięci

Algorytm FIFO zastępowania stron

Przyjrzyjmy się teraz, co się stanie, gdy liczba ramek wyniesie 4

Chwila	1	2	3	4	5	6	7	8	9	10	11	12
Odwołanie	1	2	3	4	1	2	5	1	2	3	4	5
Ramka 1	1	1	1	1	1	1	5	5	5	5	4	4
Ramka 2		2	2	2	2	2	2	1	1	1	1	5
Ramka 3			3	3	3	3	3	3	2	2	2	2
Ramka 4				4	4	4	4	4	4	3	3	3

ofiara staje się strona, która najdłużej przebywa w pamięci

Algorytm FIFO zastępowania stron

- Ciąg odniesień: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5.
- 3 ramki (3 strony mogą być w pamięci w tym samym czasie):

1	1	4	5
2	2	1	3
3	3	2	4

← 9 braków stron

- 4 ramki:

1	5	4
2	1	5
3	2	
4	3	

← 10 braków stron

Anomalia Belady'ego

Anomalia Belady'ego polega na tym, że współczynnik braków stron wzrasta ze wzrostem liczby wolnych ramek

Wykres braków stron dla algorytmu FIFO



Algorytm optymalny

Chwila	1	2	3	4	5	6	7	8	9	10	11	12
Odwołanie	1	2	3	4	1	2	5	1	2	3	4	5
Ramka 1	1	1	1	1	1	1	1	1	1	1	4	4
Ramka 2		2	2	2	2	2	2	2	2	2	2	2
Ramka 3			3	3	3	3	3	3	3	3	3	3
Ramka 4				4	4	4	5	5	5	5	5	5

ofiara staje się strona, która będzie nieużywana przez najdłuższy okres

Algorytm optymalny

□ Przykład: 4 ramki, ciąg odniesień: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5;



← 6 braków stron

- Algorytm optymalny ma najniższy współczynnik braków stron ze wszystkich algorytmów, wolny od anomalii Belady'ego
- Jest trudny w realizacji, ponieważ wymaga wiedzy o przyszłej postaci ciągu odniesień (skąd ją wziąć?).
- Jest używany głównie w studiach porównawczych jako punkt odniesienia do innych algorytmów

Algorytm LRU

Chwila	1	2	3	4	5	6	7	8	9	10	11	12
Odwołanie	1	2	3	4	1	2	5	1	2	3	4	5
Ramka 1	1	1	1	1	1	1	1	1	1	1	1	5
Ramka 2		2	2	2	2	2	2	2	2	2	2	2
Ramka 3			3	3	3	3	5	5	5	5	4	4
Ramka 4				4	4	4	4	4	4	3	3	3

ofiara strona, która nie była używana od najdłuższego czasu

Algorytm LRU

❑ Przykład: 4 ramki, ciąg odniesień: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5;

1			5
2			
3	5	4	
4	3		

← 8 braków stron

- Lepszy od algorytmu FIFO (mniej braków stron).
- Wolny od anomalii Belady'ego - często stosowany.
- Trudność z zapamiętaniem historii użycia stron – może wymagać sporego zaplecza sprzętowego.

Implementacja algorytmu LRU

- **Za pomocą liczników** - każda strona ma wówczas specjalne pole licznika. Gdy jakiś proces odwołuje się do strony, do licznika kopiuje się stan zegara systemowego.
- **Za pomocą stosu** - na stosie trzymamy numery stron, do których były odwołania. Odwołanie do strony powoduje przesunięcie jej numeru na wierzchołek tego stosu.