

Systemy operacyjne

WYKŁAD 13

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

Algorytmy przybliżające LRU

Niewiele systemów posiada odpowiedni sprzęt umożliwiający realizację algorytmu LRU

Często stosowane są algorytmy przybliżające metodę LRU:

- Algorytm bitów odniesień
- Algorytm dodatkowych bitów odniesień
- Algorytm drugiej szansy

Algorytm bitów odniesień

- Z każdą pozycją w tablicy stron związany jest bit odniesienia ustawiony początkowo na 0.
- Przy odwołaniu do strony jest on ustawiany na 1.
- Zastępowana jest ta strona w porządku FIFO, która ma bit odniesienia ustawiony na 0.

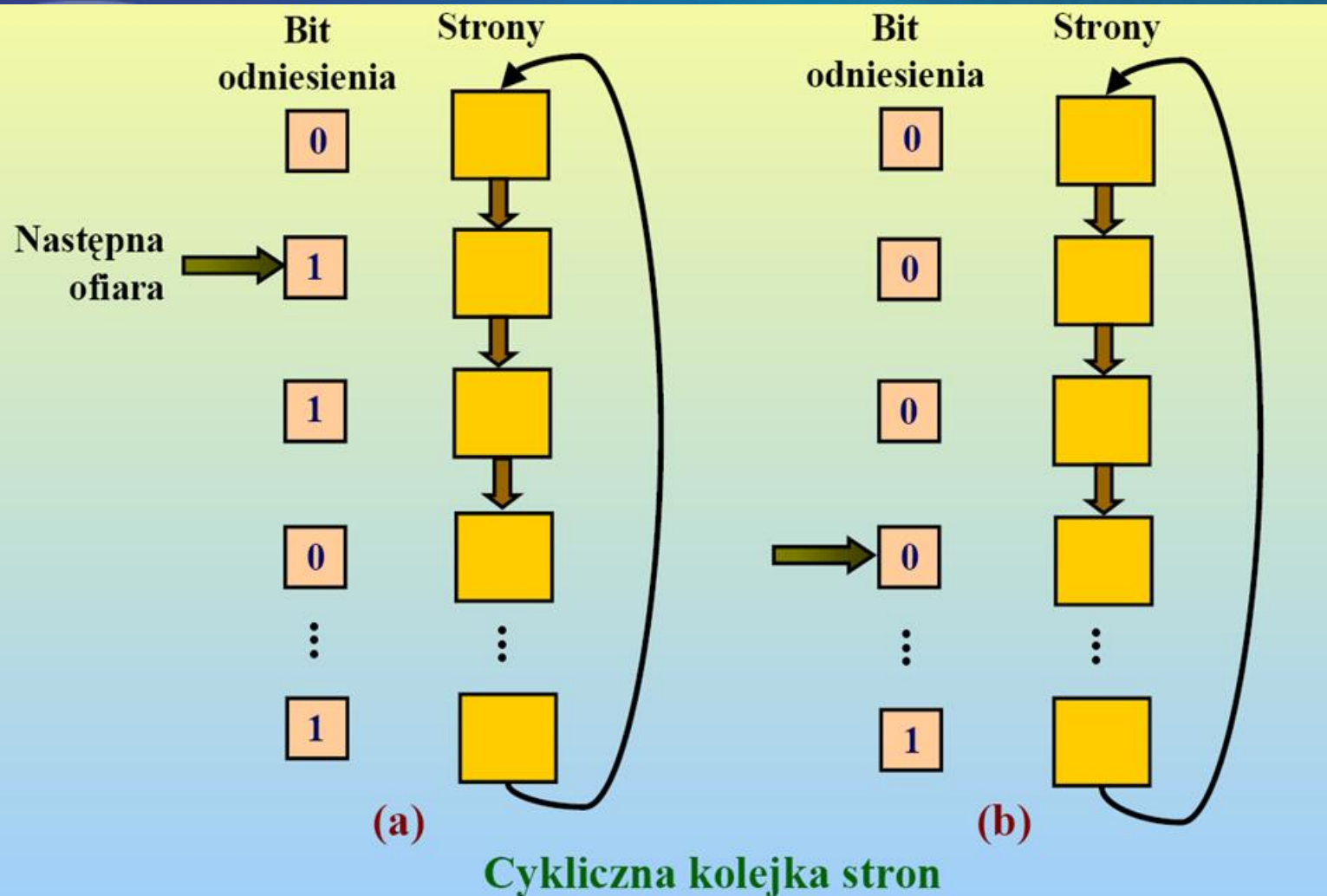
Algorytm dodatkowych bitów odniesień

- Z każdą stroną związany jest np. **8 bitowy** rejestr ustawiony na początek na **00000000**.
- W regularnych odstępach czasu (np. co 100ms) SO wprowadza na najbardziej znaczącą pozycję rejestru bit odniesienia.
- Wymieniana jest strona najdawniej używana - najmniejsza liczba w rejestrze np. **0111010 < 1100010**.
- Jeśli kilka stron ma taką samą wartość rejestru - wymiana może być realizowana na zasadach FIFO.

Algorytm drugiej szansy (zegarowy)

- Strony przeglądane są w porządku FIFO
- Sprawdzenie bitu odniesienia:
 - jeśli 0 - strona zastąpiona;
 - jeśli 1 - „druga szansa”
- „druga szansa” – zerowanie bitu odniesienia - ustawienie czasu bieżącego (koniec kolejki),
- Przewijanie stron dokonuje się cyklicznie
- Strona często eksploatowana - nigdy nie będzie zastąpiona

Algorytm drugiej szansy (zegarowy)



Ulepszony algorytm drugiej szansy

- $(0,0)$ – nieużywana ostatnio i niezmieniona – najlepsza do zastąpienia
- $(1,0)$ – używana ostatnio lecz niezmieniona – prawdopodobnie za chwilę będzie znów używana
- $(0,1)$ – nie używana ostatnio, ale zmieniona – nieco gorsza kandydatka, ponieważ stronę taką trzeba będzie zapisać na dysku przed zastąpieniem
- $(1,1)$ – używana ostatnio i zmieniona – prawdopodobnie znów będzie potrzebna, w przypadku zastępowania należy ją przekopiować z pamięci

Algorytmy zliczające

- Algorytmy zliczające korzystają ze związanego z każdą stroną licznika odwołań.
- Licznik zrealizowany sprzętowo - odwołanie do danej strony powoduje zwiększenie jej licznika o jeden.
- Istnieją dwa algorytmy oparte na przeciwnych założeniach:
 - **Least Frequently Used (LFU)** - ofiarą staje się strona, do której było najmniej odwołań (najrzadziej używana strona)
 - **Most Frequently Used (MFU)** - ofiarą staje się strona, do której było najwięcej odwołań (najczęściej używana strona).

Implementacja tych algorytmów jest kosztowna i nie przybliżają one dobrze algorytmu optymalnego

Algorytmy buforowania stron

Procedury wspomagające:

- przechowywanie puli wolnych ramek
zanim strona-ofiara zostanie usunięta potrzebna strona
czytana do wolnej ramki z puli
- przechowywanie listy zmienionych stron; zapis zmienionych
stron na dysk przez urządzenie stronicujące w wolnym czasie
- pula wolnych ramek + informacja o tym jaka strona
rezydowała w każdej ramce (możliwość ponownego jej
wykorzystania)

Algorytmy przydziału ramek

- W systemie wieloprogramowym ramki są współdzielone przez kilka procesów - procesy rywalizują o te ramki.
- Istnieje pewna minimalna liczba ramek, które muszą być przydzielone procesowi – zdefiniowana przez architekturę logiczną komputera.
- Liczba ramek musi być wystarczająca do zaalokowania wszystkich stron do których może odnosić się pojedynczy rozkaz.

Schematy przydziału ramek:

- *równy* – każdy proces dostaje tyle samo ramek.
- *proporcjonalny* – liczba ramek proporcjonalna do jego rozmiaru.
- *przydział priorytetowy* – liczba przydzielonych ramek jest proporcjonalna do priorytetu procesu lub do kombinacji rozmiaru i priorytetu.

Algorytmy przydziału ramek

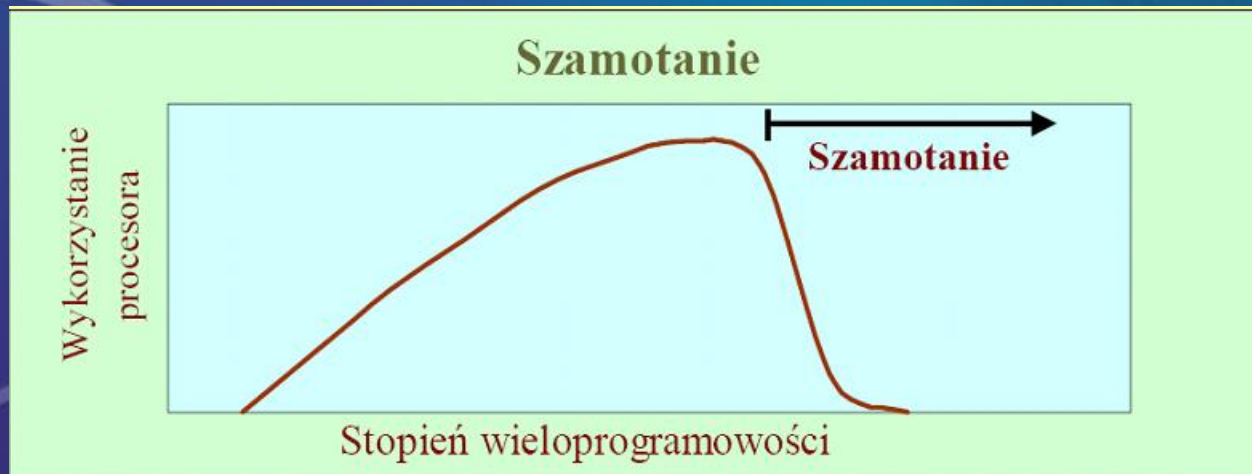
- **Zastępowanie globalne** – proces może wybierać ramki do zastąpienia ze zbioru wszystkich ramek – *jeden proces może odebrać ramkę drugiemu.*
- **Zastępowanie lokalne** – proces może wybierać ramki tylko z własnego zbioru przydzielonych ramek – *gorsza przepustowość, rzadziej stosowany.*

Szamotanie

Jeśli proces nie ma wystarczającej liczby stron w pamięci operacyjnej to częstość braków stron będzie wysoka - prowadzi to do:

- Zmniejszenia wykorzystania procesora - mniejsze wykorzystanie procesora może być sygnałem dla planisty do zwiększenia wieloprogramowości.
- Załadowanie kolejnego procesu zostaje do systemu pogarsza sytuację – zwiększa liczbę braków stron.
- Wykorzystanie procesora się zmniejsza co jest sygnałem dla planisty do dalszego zwiększania wieloprogramowości.
- Powstaje szamotanie – *przepustowość systemu gwałtownie spada*

Szamotanie



szamotanie to zmniejszenie wykorzystania procesora przy zwiększeniu stopnia wieloprogramowości (wykonywanie przesyłania pomiędzy pamięcią operacyjną a dyskiem)

Szamotanie

- Aby zapobiec szamotaniu należy dostarczyć procesowi tyle ramek ile potrzebuje – ***jak się o tym dowiedzieć?***
- **Aby ograniczyć szamotanie należy:**
 - Zastosować lokalny lub priorytetowy algorytm zastępowania stron.
 - Dostarczyć procesowi właściwą liczbę ramek - strategia tworzenia zbioru roboczego **model strefowy wykonania procesu.**