# Pamięć wirtualna

# Kolejki OS

- **Długookresowa** – dla procesów oczekujących na wejście do systemu
- **Krótkookresowa** – dla procesów działających w systemie, gotowych do operacji na procesorze
- **We/Wy** – skojarzona z każdym urządzeniem we/wy, dla procesów oczekujących na nie

- **Fakt**: większość procesów czeka na we/wy
- **Ergo**: procesor się nudzi
- **Zatem**: dużo procesów ma szansę działania
- **Ale**:
  - dużo procesów → dużo pamięci
  - przełączanie CPU między procesami
  - nadzór nad dostępem

# Dużo procesów – dajmy im pamięć

- Cena pamięci
- Zapotrzebowanie na pamięć wciąż rośnie
- Paradoks: *im więcej jest dostępnej pamięci, tym bardziej pamięciożerne stają się procesy*

# Dużo procesów – wymieniajmy je

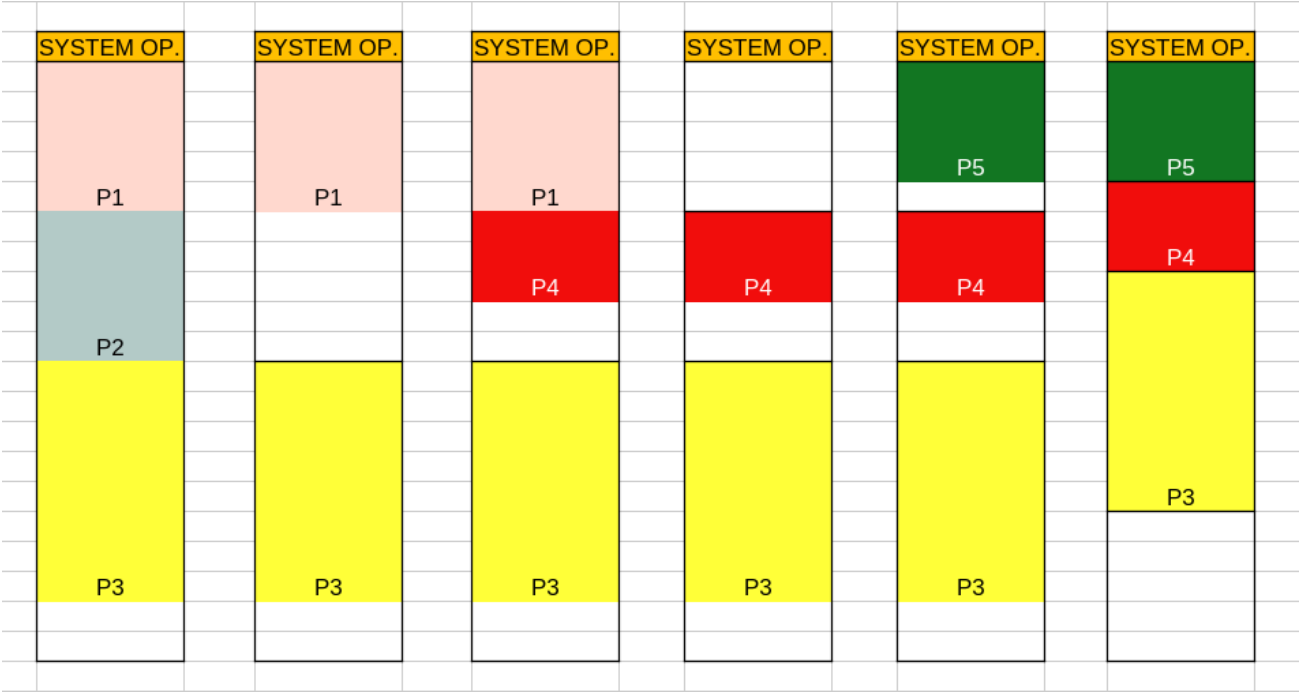
1) Proces czekający na We/Wy odesłany na dysk (kolejka We/Wy, potem kolejka krótkookr.)
2) Wchodzi inny proces (z kolejki długookr. lub krótkookr.)
3) Proces zakończony jest usuwany z pamięci

To są wszystko operacje We/Wy…

# Partycjonowanie / Segmentacja

- Proces dostaje „swoją” część pamięci – segment
- Rozmiar: stały (kilka rozmiarów) lub zmienny
- Gdy zaczyna brakować – jak poprzednio
- Powstają „nieużytki” – dziury w pamięci
  - upakowanie
  - zrzucenie na dysk kilku procesów (których?!)

# Przykład fragmentacji



# Powstaje coś nowego!

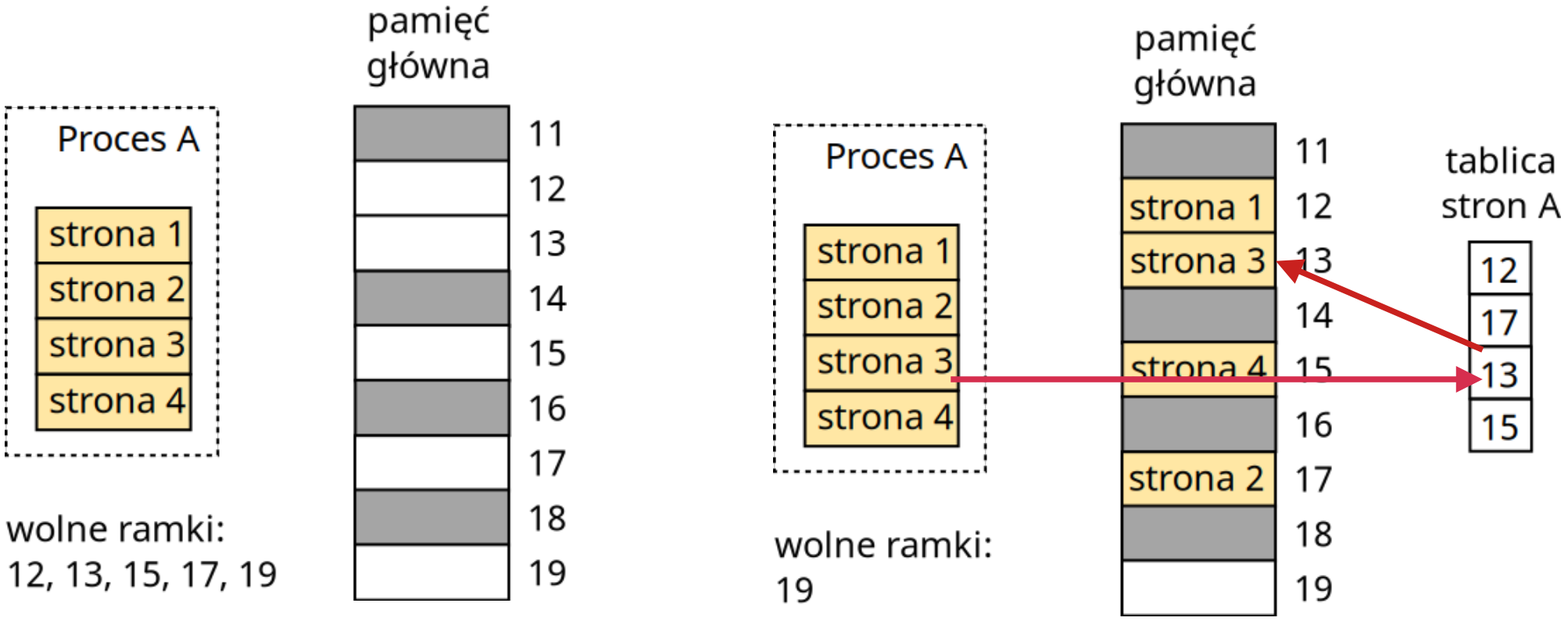
- Przeładowywanie i upakowywanie powoduje zmianę adresów *fizycznych* (lokacja danych, adresy skoków) widzianych przez program!
- Nowe adresy:
  - logiczny – względem początku programu
  - fizyczny – względem początku pamięci
  - bazowy – np. początek programu (albo czegoś innego)

$$Af = Ab + Al$$

# Stronicowanie

- Zmniejszamy rozmiar partycji (to są *ramki*)
- Dzielimy programy na części o takim samym rozmiarze (to są *strony*)
- Ładując program, ładujemy strony do ramek (niekoniecznie kolejnych)
- Wszystko trzymamy w **tablicy stron** (mapa)
- Mniejszy program – mniej stron/ramek
  - fragmentacja tylko na ostatniej ramce

# Przykład ładowania w stronicowaniu



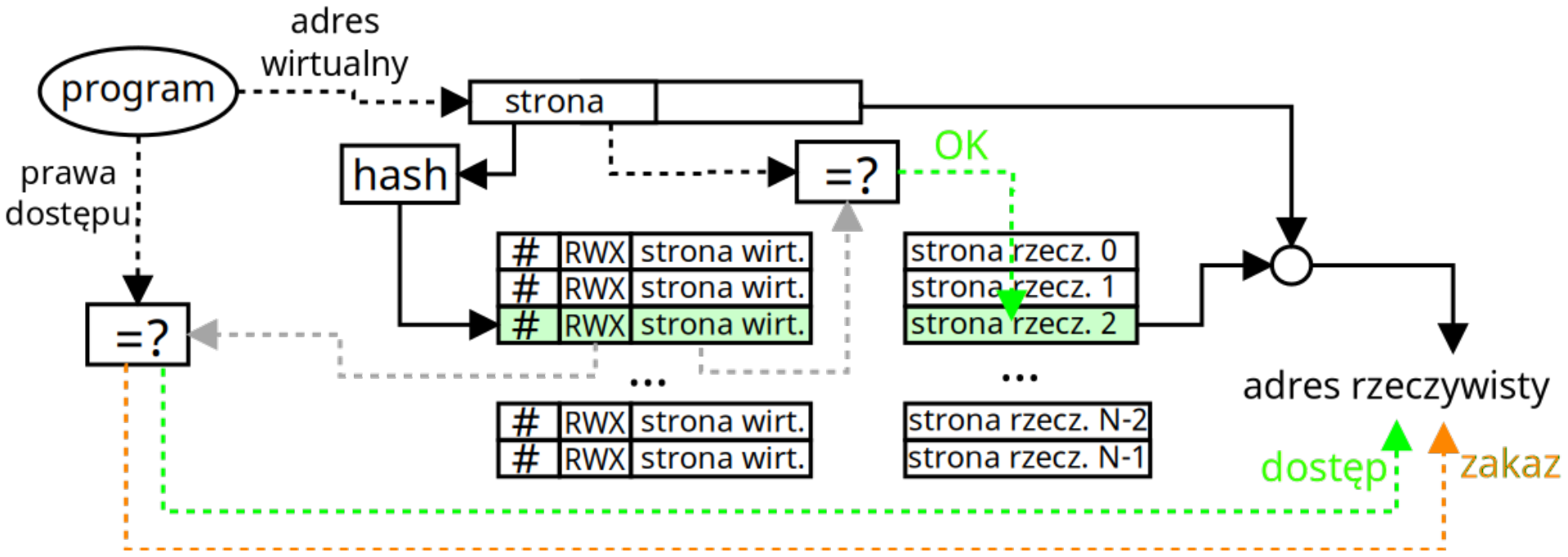
# Stronicowanie na żądanie

- Nie trzeba od razu ładować wszystkich stron programu (zasada lokalności…)
- Jeśli nie ma żądanej strony:
  - błąd strony → fetch
  - uwaga na szamotanie
  - algorytmy wymiany?
- Skoro „nieużywane” słowa są na dysku, to…
  - proces może być większy, niż dostępna pamięć
- Pamięć **rzeczywista** vs. **wirtualna**

# Bałagan w tablicach stron

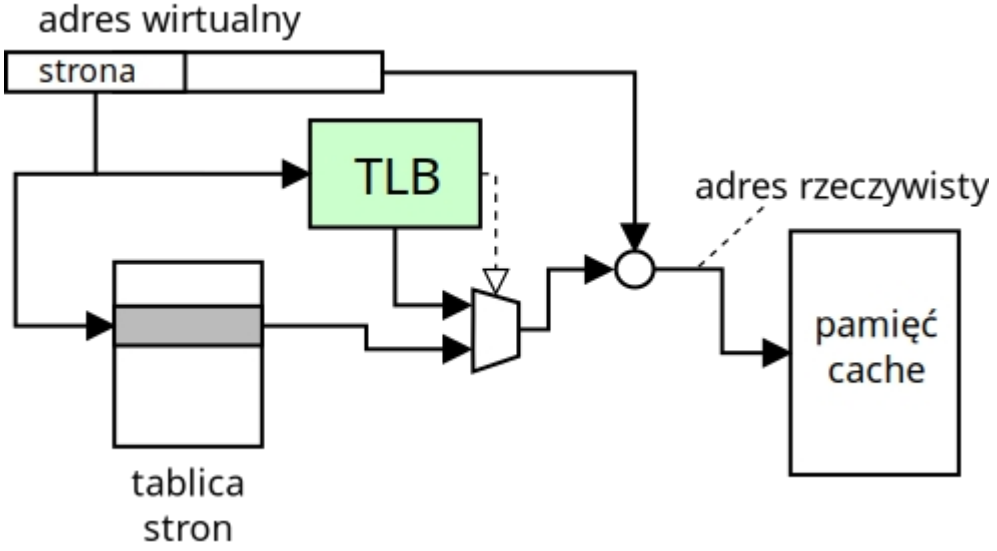
- Typowo: jedna na proces, ale wiele procesów!
- Proces ma 2 GB, strona 512 B, tablica ma…?
- Tablice stron przechowywane w pam. wirtualnej
  - … a więc też są stronicowane – podwójny dostęp
- Razem z procesem tablica w pam. głównej
  - dodatkowa pamięć do przeładowania
- W x86 jest katalog tablic (dwupoziomowo)
- Odwrócona tablica stron

# Odwrócona tablica stron



# Bufor translacji adresów stron (TLB)

- Żeby nie zaglądać ciągle do tablicy stron

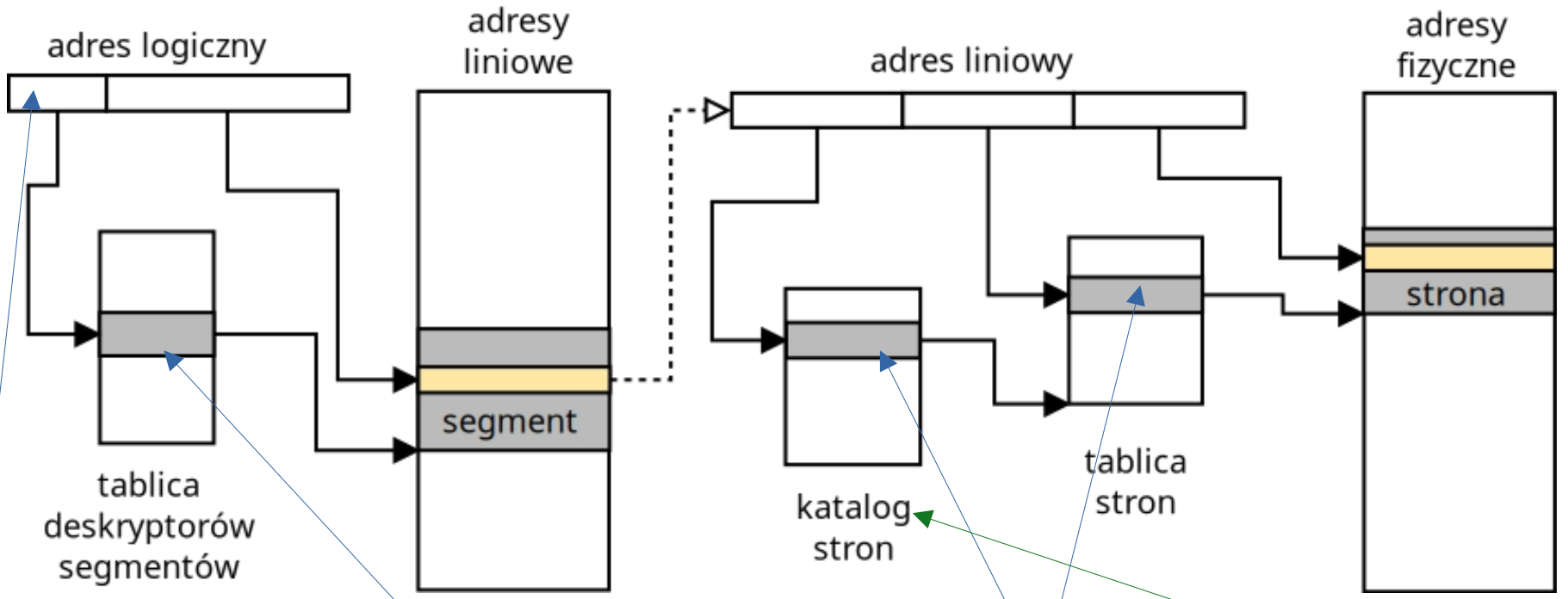


# Segmentacja

- Pod kontrolą programisty
- Umożliwia podział przestrzeni adresowej na obszary zawierające określone dane, dostęp, prawo wykonania, właściciela...
- Możliwość współdzielenia
- Segmenty można niezależnie modyfikować i ładować

- Segmentacja stronicowana

# Segmentacja stronicowana




poziom uprzywilejowania
segment globalny/lokalny
obecny w pamięci rzecz.?

rozmiar segmentu
RWX, code, data

R/W
user/supervisor
cache block
obecny w pamięci rzecz.?

zawsze w pamięci głównej
ze swoim procesem