

# Systemy operacyjne

## WYKŁAD 15

dr inż. Stanisława Plichta

[splichta@ans-ns.edu.pl](mailto:splichta@ans-ns.edu.pl)

# Obsługa urządzeń wejścia/wyjścia

## Rozważmy operację odczytu pliku z dysku

- Ustalenie urządzenia, na którym znajduje się ten plik
- Przetłumaczenia nazwy tego urządzenia na wewnętrzny identyfikator.
- Fizyczny odczyt danych z dysku do bufora.
- Udostępnienie danych zlecającemu procesowi.
- Przekazanie sterowania temu procesowi.

# Obsługa urządzeń wejścia/wyjścia

- **Proces użytkownika**: zlecam odczyt bloku
- **Jądro**: sprawdzam, czy można już zrealizować zlecenie (np. dzięki **pamięci podręcznej**)
- **Jądro**: jeśli tak, to udostępniam wynik zlecenia. Jeśli nie, to wysyłam zlecenie do modułu sterującego urządzeniem
- **Moduł sterujący**: wydaję polecenia sterownikowi urządzenia
- **Sterownik**: steruję urządzeniem, przerywam po zakończeniu operacji wejścia/wyjścia
- **Sterownik**: generuję **przerwanie**
- **Procedura obsługi przerwania**: składam dane w buforze.
- **Moduł sterujący**: ustaliam, którą operację wejścia/wyjścia zakończono, informuję jądro o zmianie stanu operacji
- **Jądro**: przekazuję dane procesowi użytkownika
- **Proces użytkownika**: mam dane lub wiem, że był błąd

# Wydajność

- Wejście/wyjście ma szczególnie duży wpływ na wydajność systemu operacyjnego, ponieważ:
  - Wymaga zasobów procesora do wykonywania modułu sterującego i kodu podsystemy wejścia/wyjścia w jądrze.
  - Powoduje przełączenia kontekstu w związku z obsługą przerwań (albo aktywne czekanie przy programowanym wejściu/wyjściu),
  - Oznacza kopiowanie danych.

# Wydajność

## Wydajność - użytkownik wciska klawisz

- Sterownik klawiatury generuje przerwanie.
- Aktywują się kolejno moduł sterujący klawiatury i podsystem jądra.
- Następuje przełączenie kontekstu na proces użytkownika, który zleca wysłanie wpisanego znaku przez sieć do zdalnej maszyny.
- konstruowany jest pakiet sieciowy i przekazywany do modułu obsługi karty sieciowej.
- Wszystko musi przejść znów przez obsługę przerwania i przełączenie kontekstu.
- Pakiet jest odbierany na maszynie zdalnej, gdzie powoduje przerwanie i obsługę.
- Z pakietu wyjmuje się wpisany znak i przekazuje do procesu aplikacyjnego, który ma ten znak zrozumieć i obsłużyć.



# Poprawa wydajności

Poprawę wydajności wejścia/wyjścia osiągniemy przez:

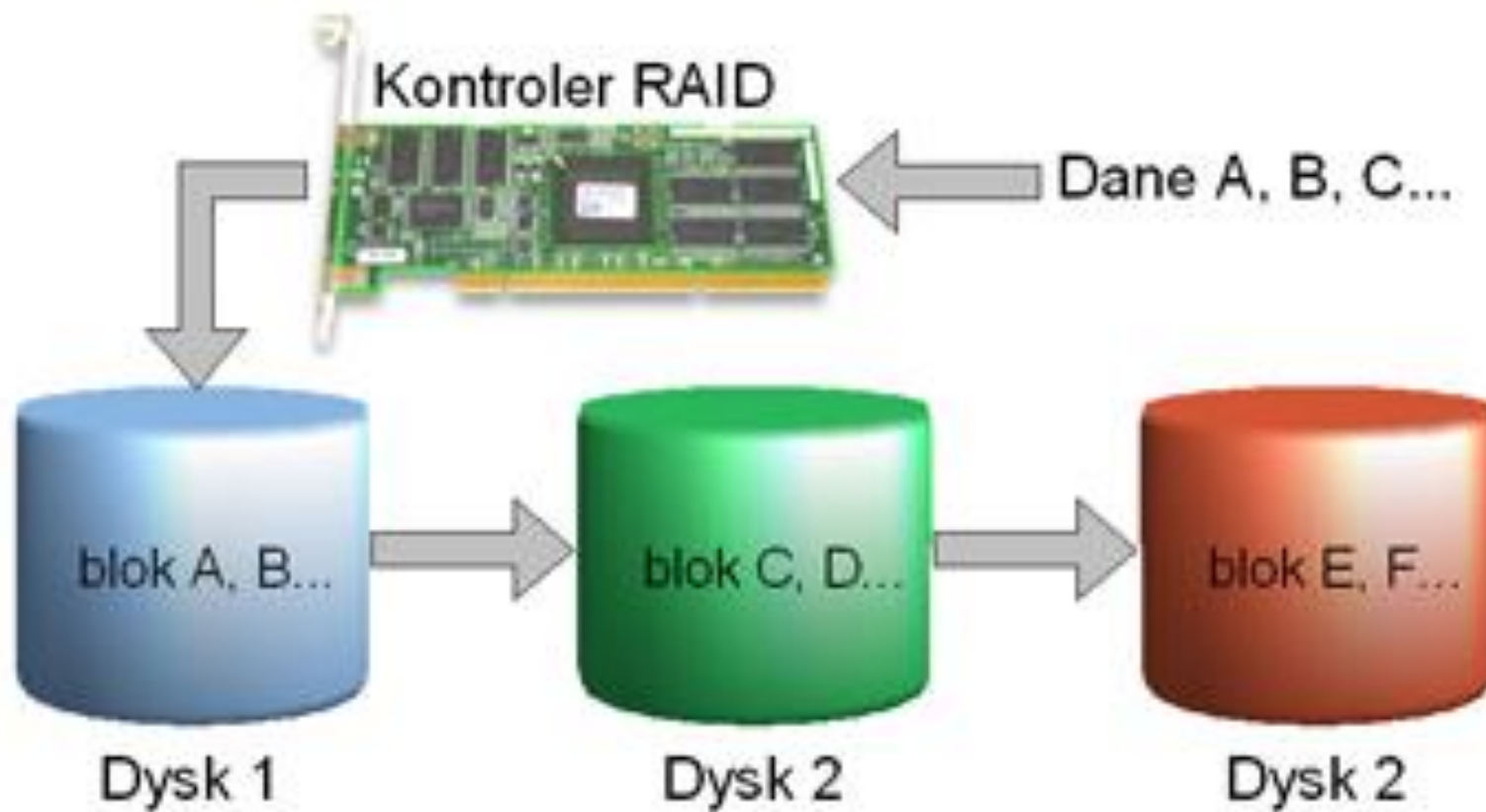
- zmniejszenie liczby przełączeń kontekstu,
- zmniejszenie zakresu kopiowanie danych,
- ograniczenie liczby przerwania poprzez wykonywanie operacji wejścia/wyjścia na większych blokach danych, wykorzystanie inteligentnych sterowników i rozsądne wykorzystanie odpytywanie,
- wykorzystanie bezpośredni dostęp do pamięci
- zwiększenie równoległości poprzez wykonywanie elementarnych działań za pomocą sprzętu w sterownikach urządzeń,
- zrównoważenie wydajności procesora, pamięci, szyn i operacji wejścia wyjścia.

# Realizacja funkcji we/wy

Funkcje urządzeń wejścia/wyjścia można zrealizować w następujących miejscach:

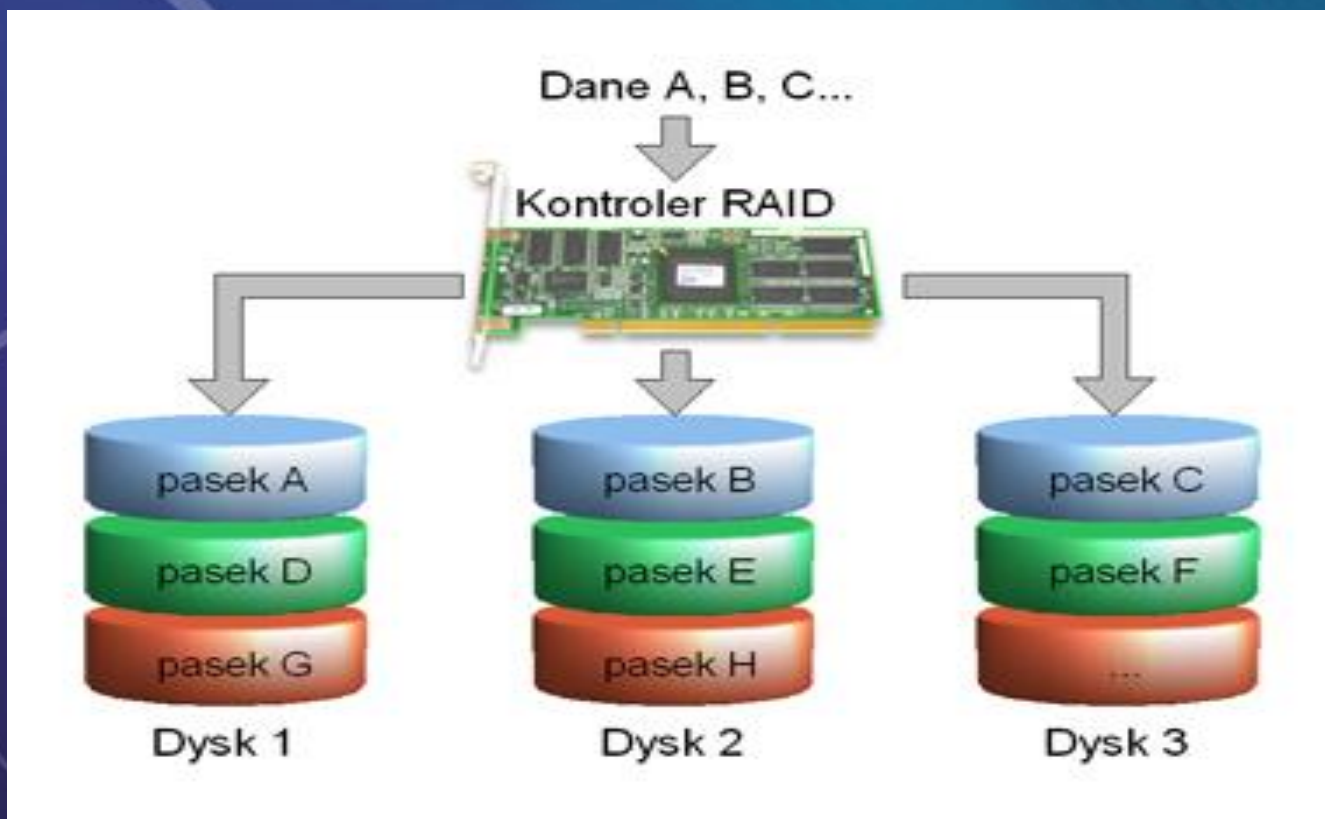
- W kodzie aplikacji (oprogramowanie).
- W kodzie jądra (oprogramowanie).
- W kodzie modułu sterującego (oprogramowanie).
- W kodzie sterownika (sprzęt).
- W kodzie urządzenia (sprzęt).

# Macierze dyskowe

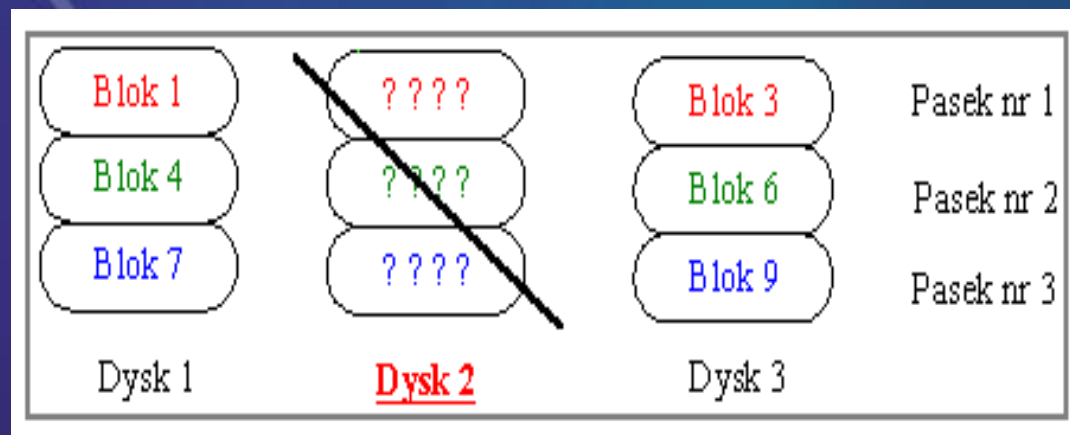
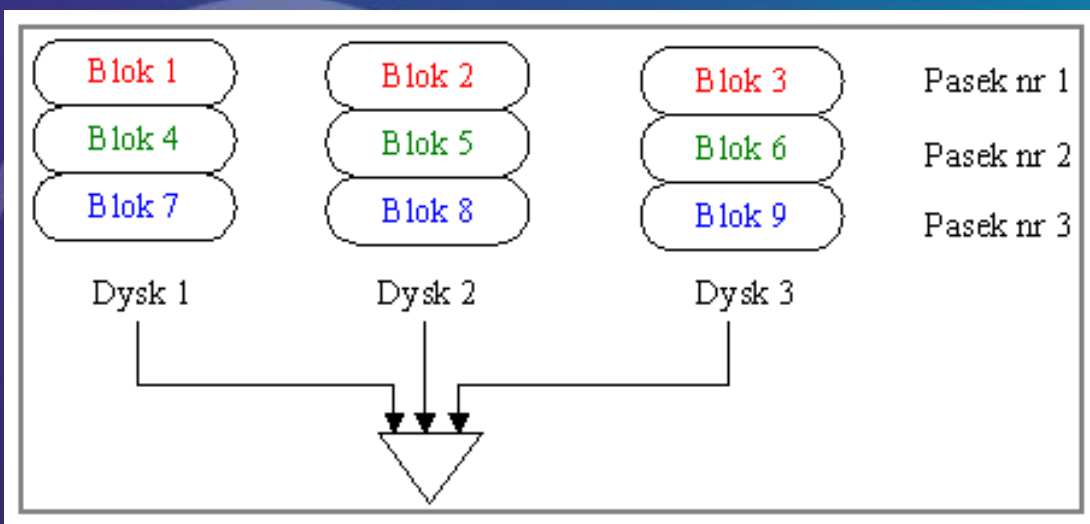




# Architektura RAID 0



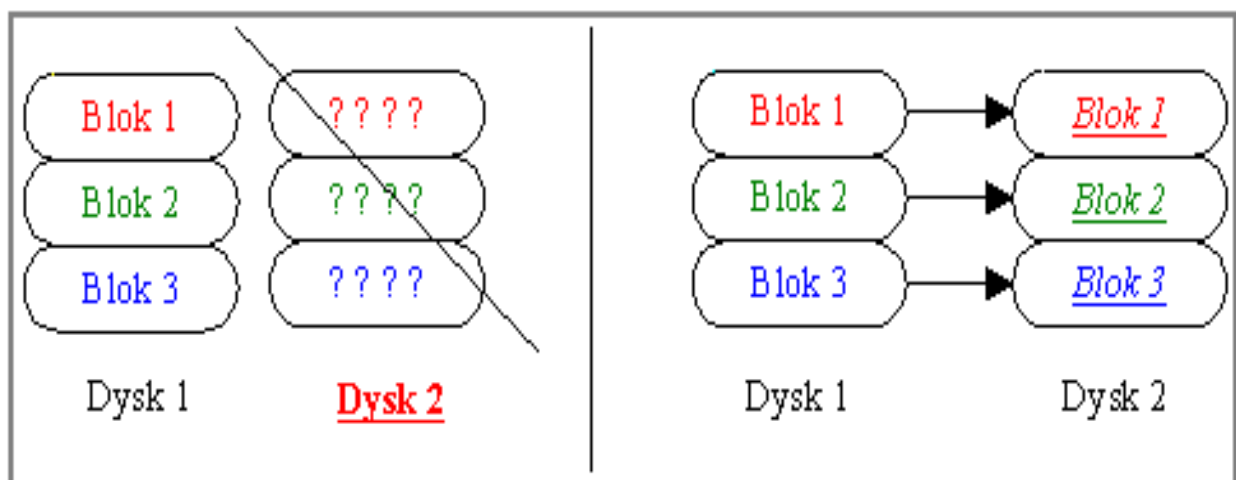
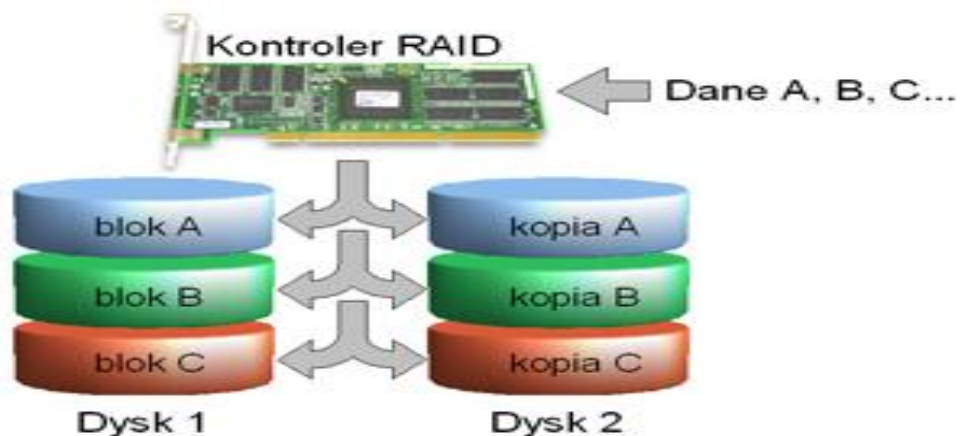
# Architektura RAID 0



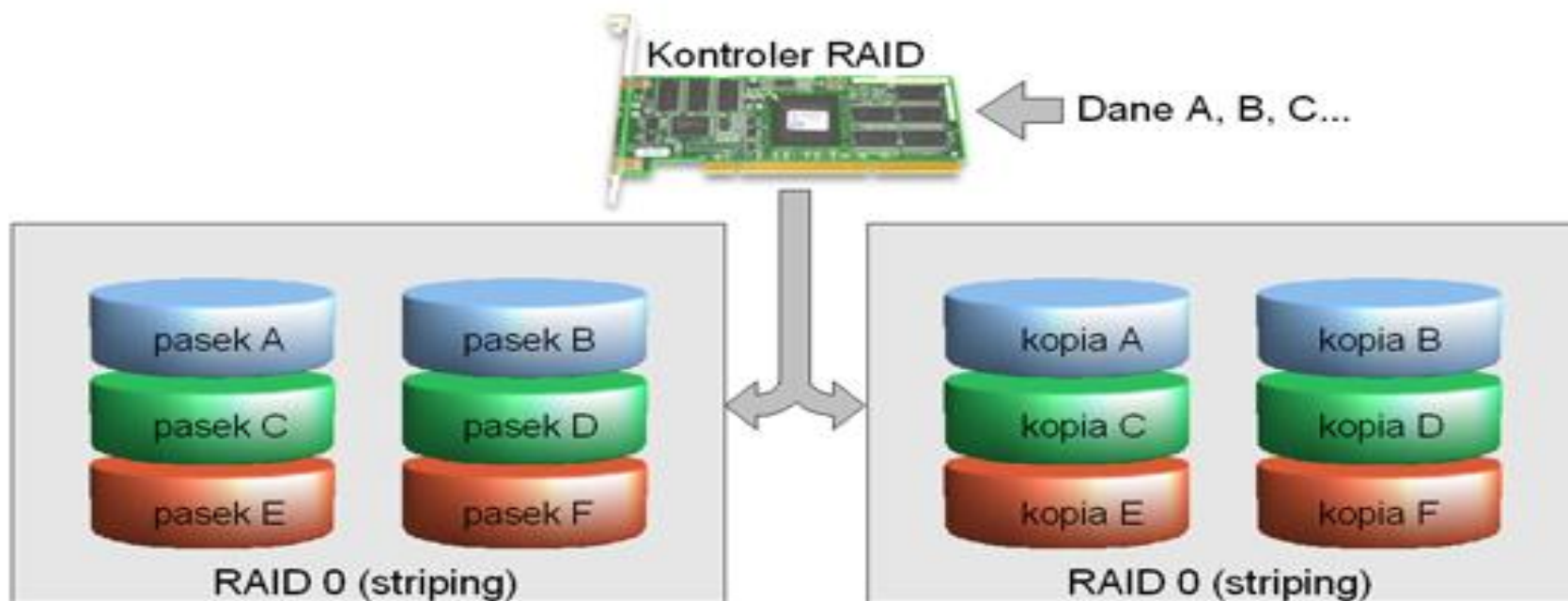
# Architektura RAID 1

- **RAID 1 (Mirroring)** – Realizowany jest zapis danych na dyskach połączonych w pary, dane na obu dyskach w każdej parze są identyczne.
- Zaletą jest wysoki poziom bezpieczeństwa
- wadą - utrata połowy pojemności dysków.
- RAID 1 jest jedynym poziomem RAID, który może zapewnić bezpieczeństwo, wykorzystując jedynie 2 dyski, pozostałe wymagają przynajmniej trzech.

# Architektura RAID 1



# Architektura RAID 0 + 1

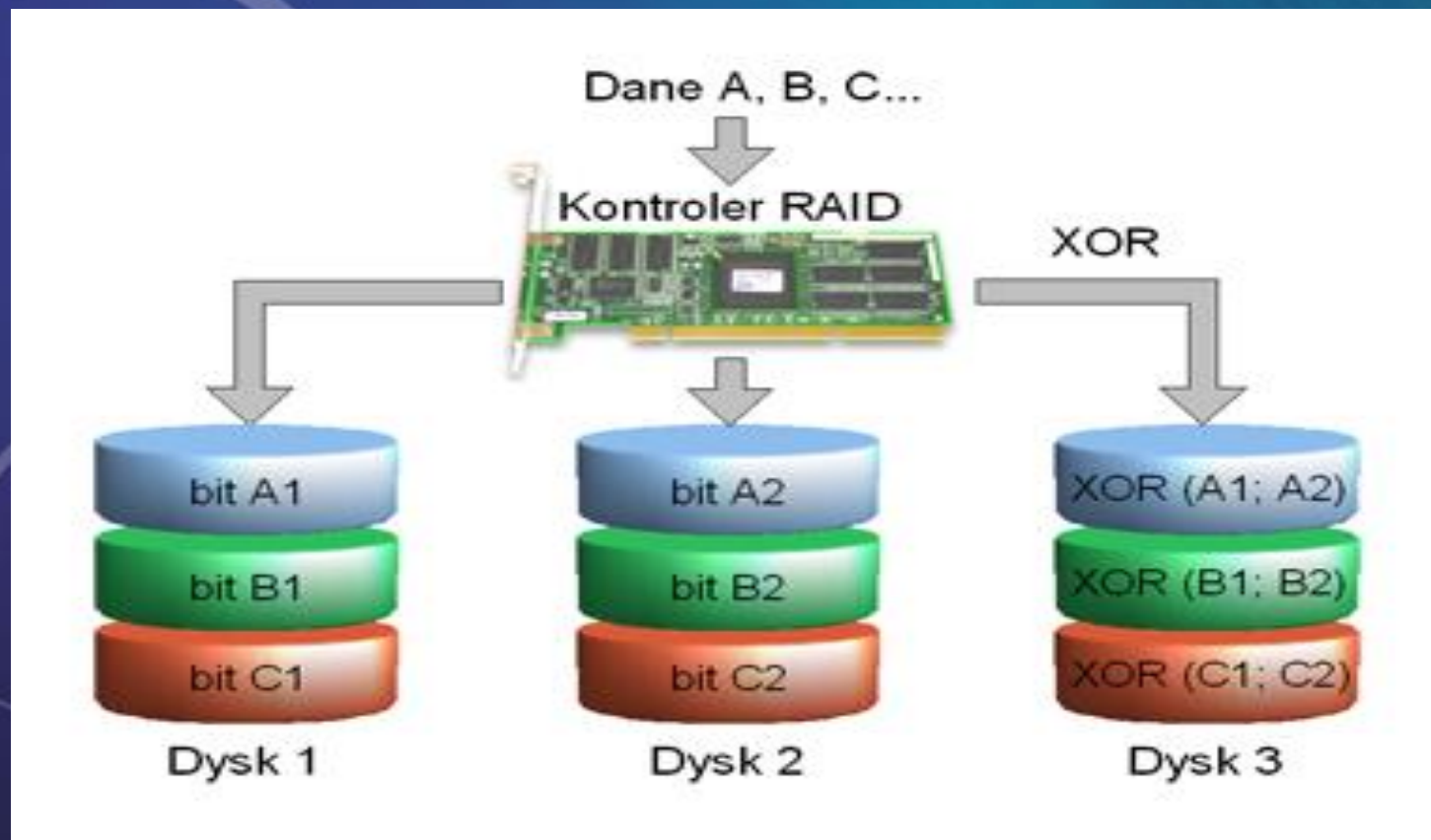




# Architektura RAID 3

- **RAID 3** - dane są zapisywane na kilku dyskach z tym, że jeden dysk jest dyskiem parzystości.
- Wydajność macierzy jest gorsza niż w RAID 0, ale wykorzystywana jest większa część przestrzeni dyskowej niż w RAID 1.
- Wadą RAID 3 jest to, że dyski są nierównomiernie obciążone (dysk zapisujący informacje o parzystości jest obciążony znacznie bardziej od pozostałych).

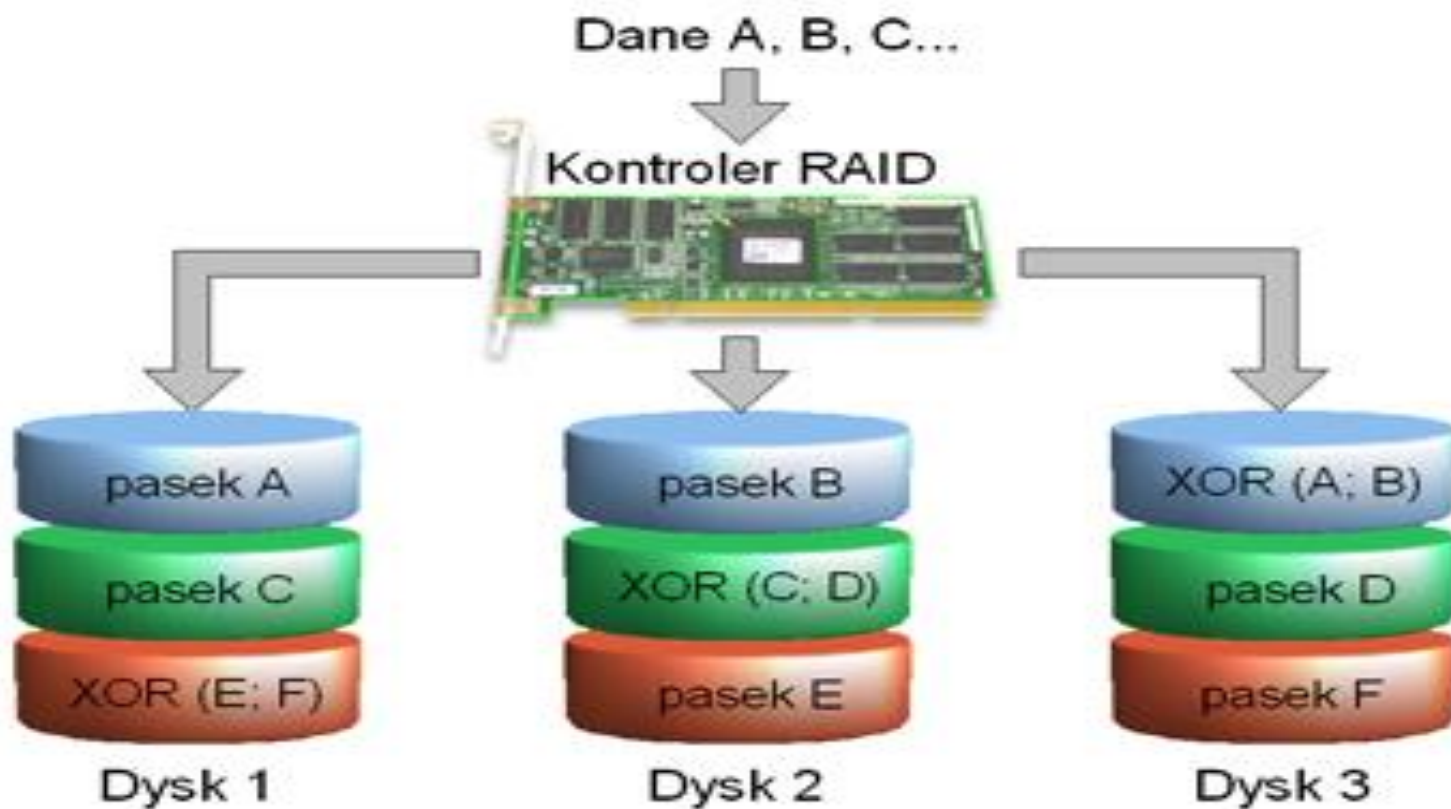
# Architektura RAID 3



# Architektura RAID 5

- **RAID 5** - Zapis danych jest realizowany na kilku dyskach jednocześnie, jak w RAID 3, ale z "rotacyjnym" zapisem kodów parzystości na wszystkich dyskach pracujących w macierzy.
- Dzięki temu dyski są równomiernie obciążone, zaś odczyt (szczególnie przy pracy losowej) odbywa się szybciej niż w RAID 3.
- Zaletą RAID 5 jest też bezpieczeństwo danych i efektywne wykorzystanie powierzchni nośnika, dlatego jest to najczęściej stosowany poziom RAID.

# Architektura RAID 5

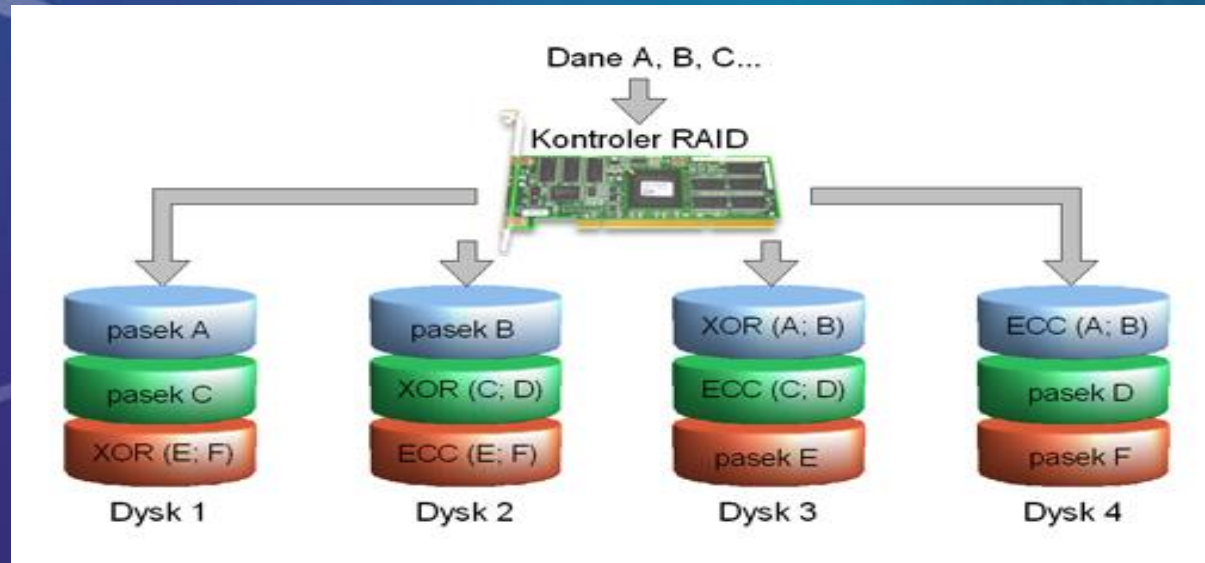


# Architektura RAID 5

- Współbieżnie, ze wszystkich dysków należących do danego paska, odczytuje się zawartości wszystkich bloków
- Wylicza się nową wartość bloku parzystości
- Parzystość wraz z zawartością modyfikowanego bloku oraz dodatkowo identyfikator (numer) paska, którego dotyczy ta informacja, zapisuje się w specjalnym obszarze systemowym (log area).
- Aby zachować spójność cały proces realizuje tylko jedna operacja wejścia-wyjścia. Te dane umożliwiają po ewentualnej awarii, przeprowadzenie procesu uaktualnienia zawartości paska.
- Modyfikuje się zawartość macierzy - dane oraz parzystość. Zapis na dwóch dyskach przebiega współbieżnie.
- Po zakończeniu operacji unieważnia się informacje zapisane w obszarze systemowym



# Architektura RAID 6



- RAID 6 – to schemat nadmiarowości.
- Technika ta przypomina RAID poziomu 5, lecz przechowuje się w niej dodatkową informację nadmiarową na wypadek zwielowrotnionych awarii dysków.
- Zamiast wykorzystywania parzystości stosuje się kody korygujące.
- Na każde 4 bity danych pamięta się 2 bity danych nadmiarowych.