

# Systemy operacyjne

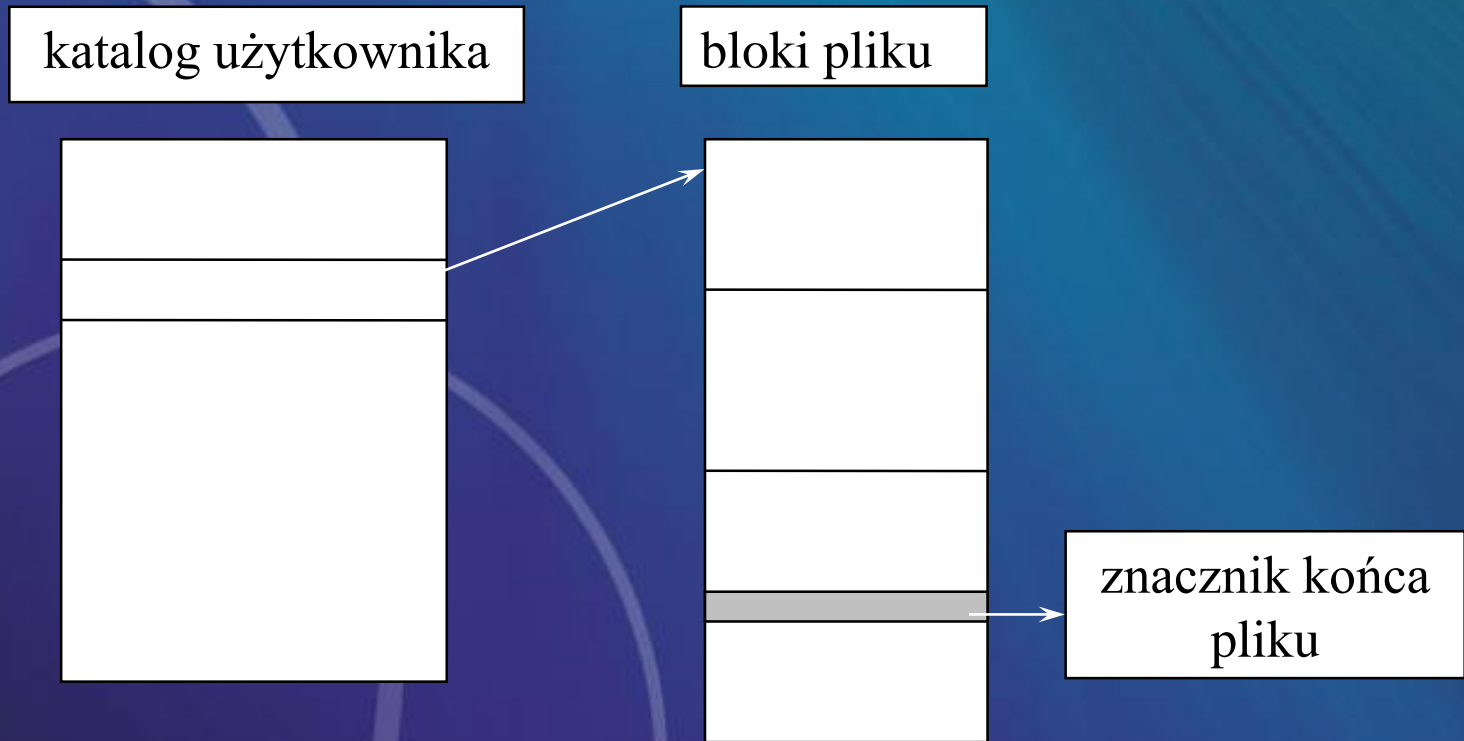
## WYKŁAD 4

dr inż. Stanisława Plichta  
[splichta@ans-ns.edu.pl](mailto:splichta@ans-ns.edu.pl)

# Metody przydziału miejsca na dysku

- System plików zwartych (przydział ciągły).
- Łańcuch powiązanych bloków (przydział listowy).
- Mapa plików (tablica alokacji).
- Bloki indeksów.

# Metody przydziału miejsca na dysku



# Metody przydziału miejsca na dysku



**blok początkowy: 1**  
**rozmiar: 4**

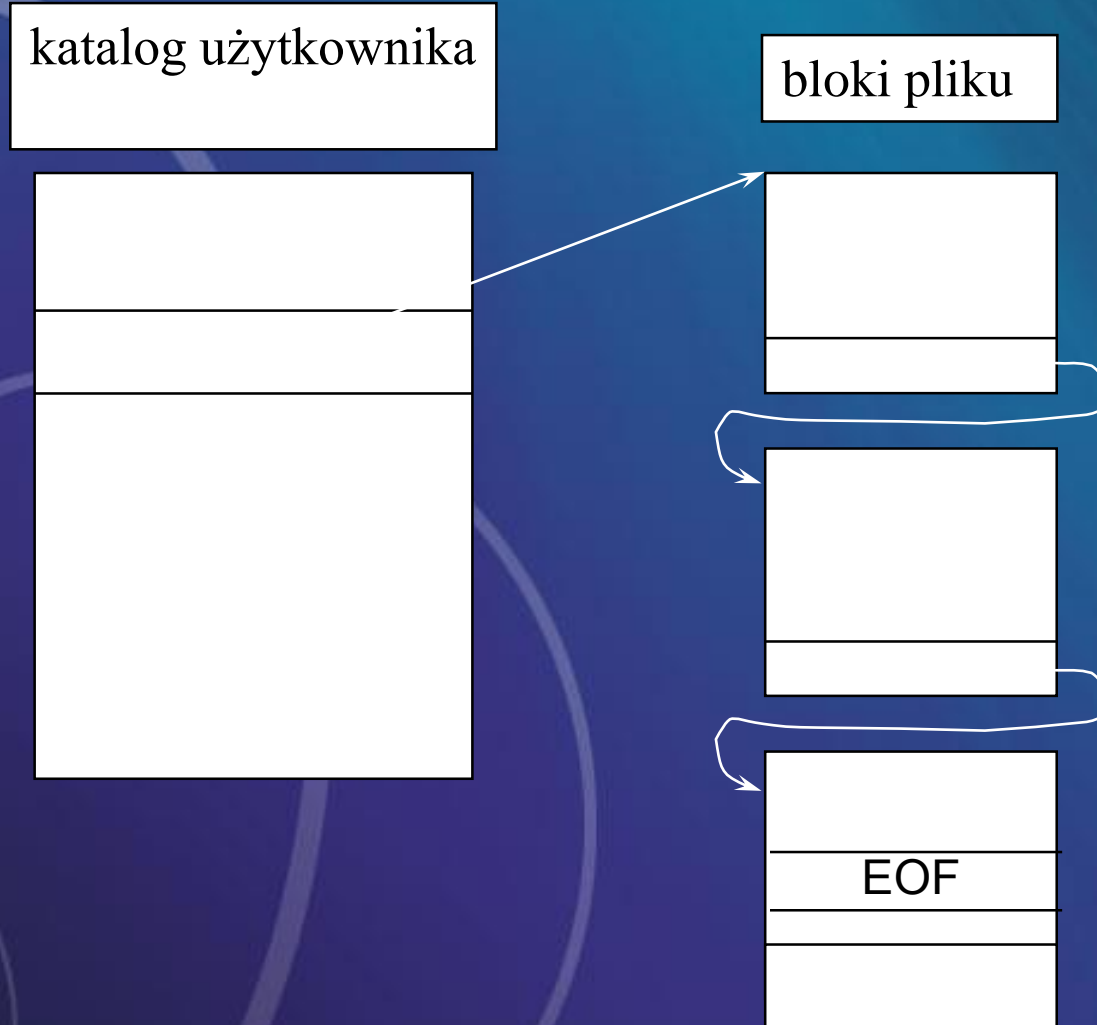
**blok początkowy: 10**  
**rozmiar: 5**

**blok początkowy: 20**  
**rozmiar: 2**

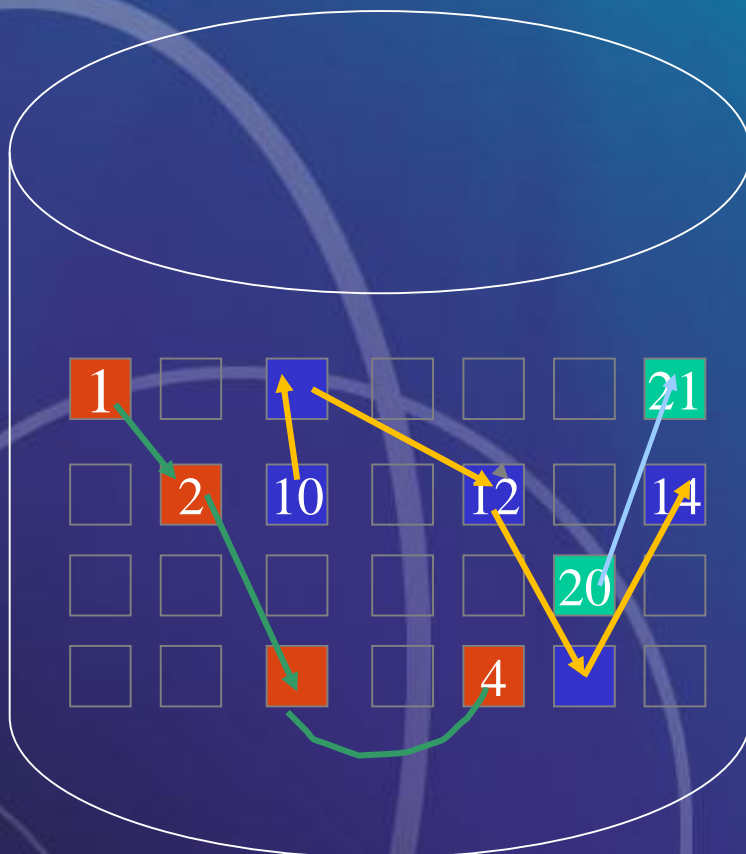
# Właściwości - zalety i wady

- Łatwo implementować dostęp swobodny i sekwencyjny.
- Trudno uniknąć fragmentacji zewnętrznej.
- Umożliwia najbardziej elastyczną organizację danych - zniszczenie jednego bloku powoduje tylko lokalną utratę danych.
- Odpowiednie do takich zastosowań jak bazy danych.

# Przydział listowy



# Przydział listowy



**blok początkowy: 1**  
**rozmiar: 4**

**blok początkowy: 10**  
**rozmiar: 5**

**blok początkowy: 20**  
**rozmiar: 2**



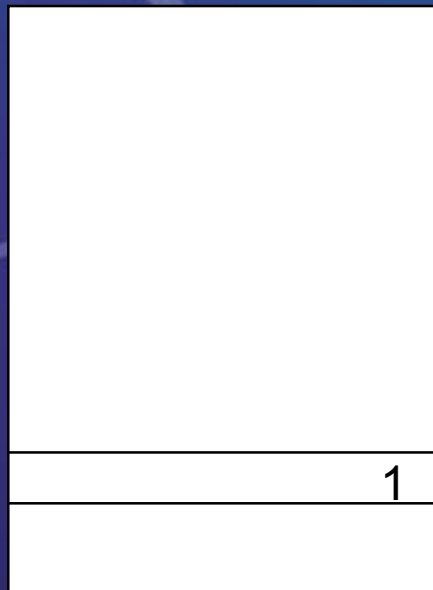
# Właściwości - zalety i wady

- Kilka bajtów, każdego bloku w pliku służy jako wskaźnik do następnego bloku.
- Wada - konieczność uzyskania dużej liczby dostępów do dysku, zanim znajdzie się koniec pliku.
- Dostęp do pliku jest z konieczności sekwencyjny.
- Metoda ta jest mało elastyczna - skutki uszkodzenia jednego bloku mogą niespodziewanie rozszerzyć się na cały system plików.
- Nie ma fragmentacji zewnętrznej.



# Mapa plików

katalog użytkownika



mapa pliku

0

1

2

3

4

5

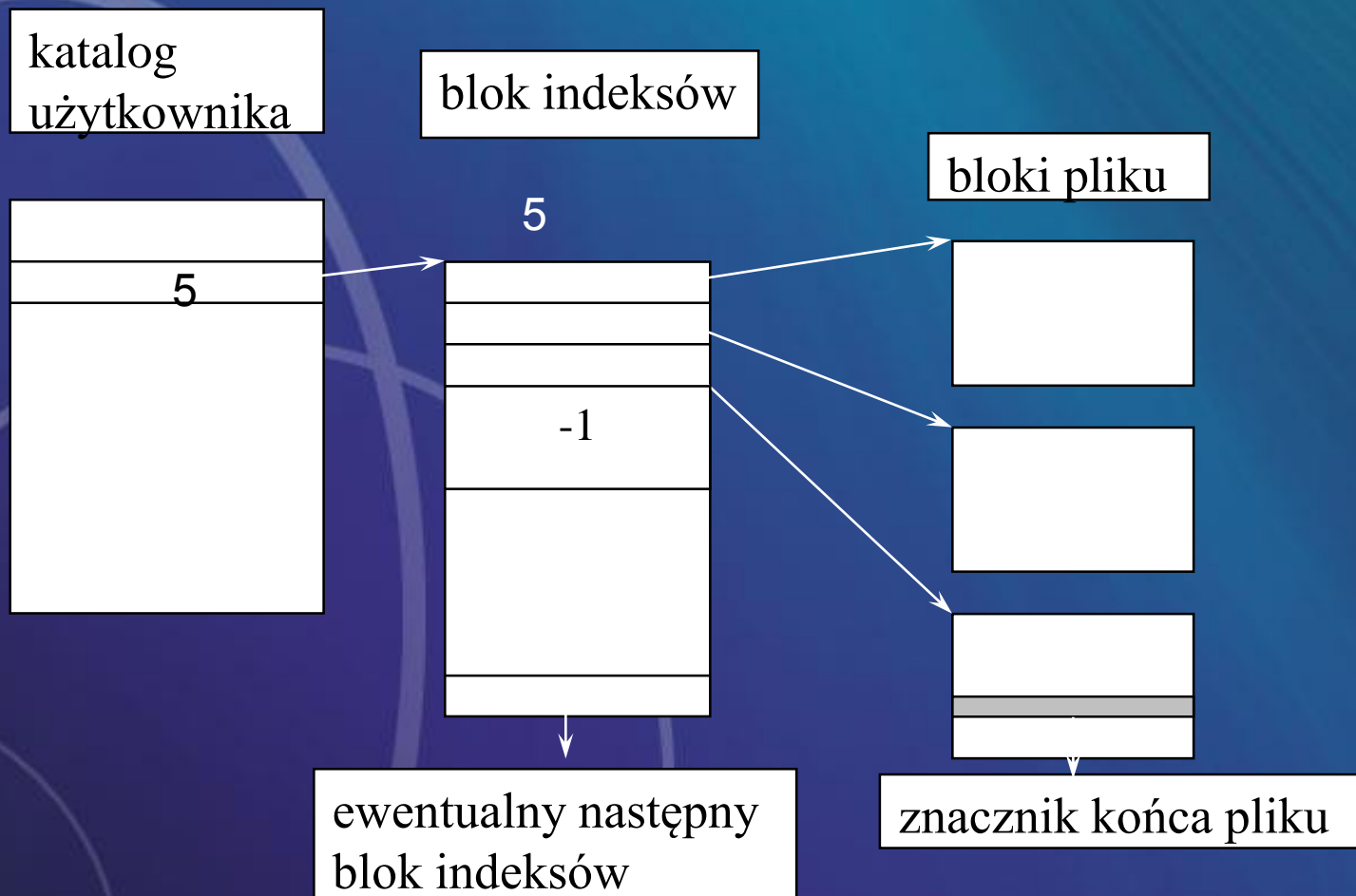


wskaźnik pusty

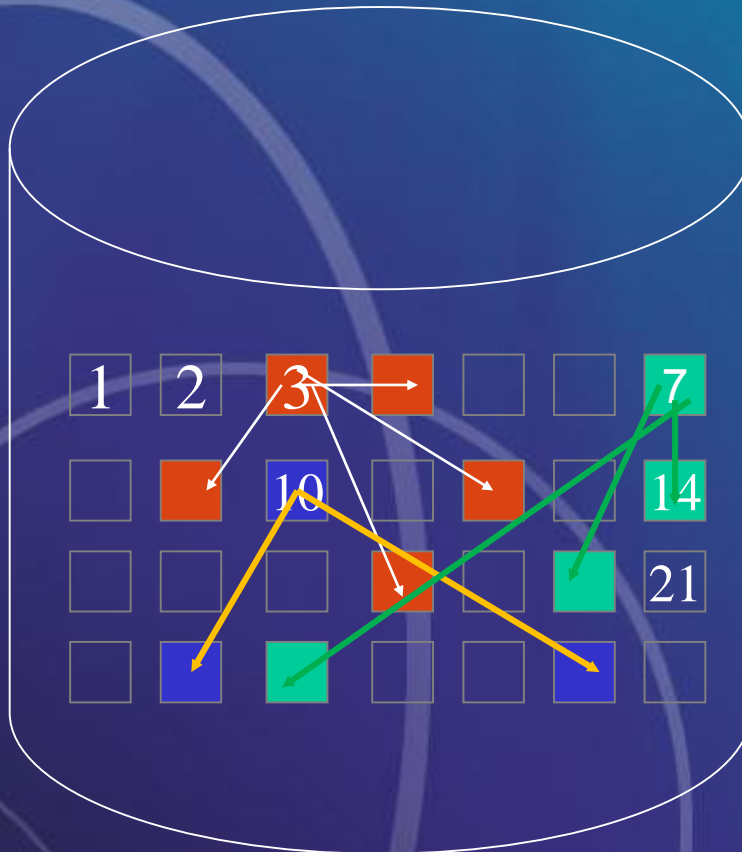
# Właściwości - zalety i wady

- Każdy blok na dysku - pozycja w mapie.
- Bloki nieużywane - 0 w tablicy.
- Uszkodzenie mapy plików może spowodować poważne straty danych - dwie kopie mapy w różnych rejonach dysku, aby w razie awarii sprzętu nie zniszczyć wszystkich kopii.
- Znaczny ruch głowic dyskowych.
- Polepszenie czasu dostępu swobodnego.

# Przydział indeksowy



# Przydział indeksowy



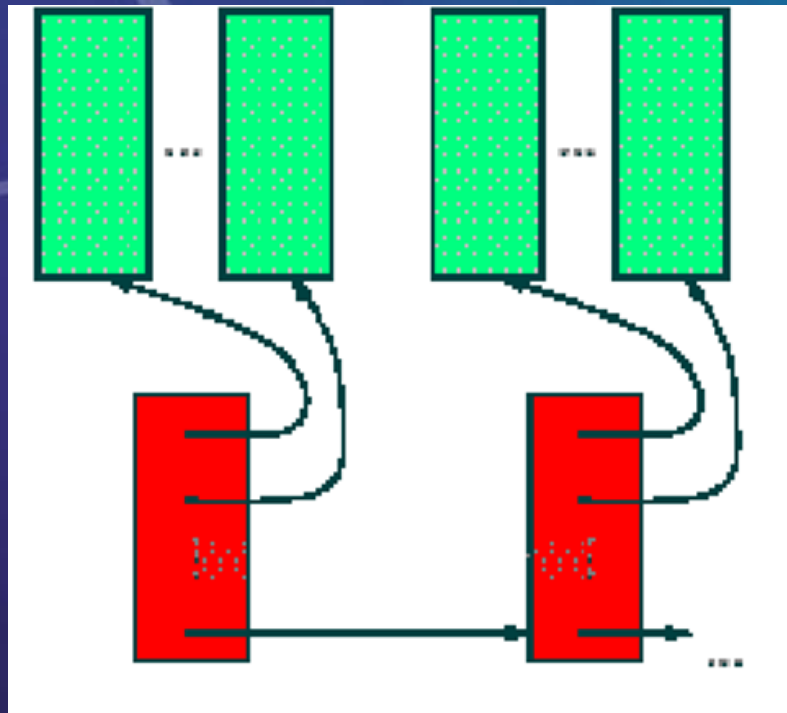
**blok indeksowy: 3**  
**rozmiar: 4 bloki**



**blok indeksowy: 10**  
**rozmiar: 2 bloki**

**blok indeksowy: 7**  
**rozmiar: 3**

# Struktura bloku indeksowego

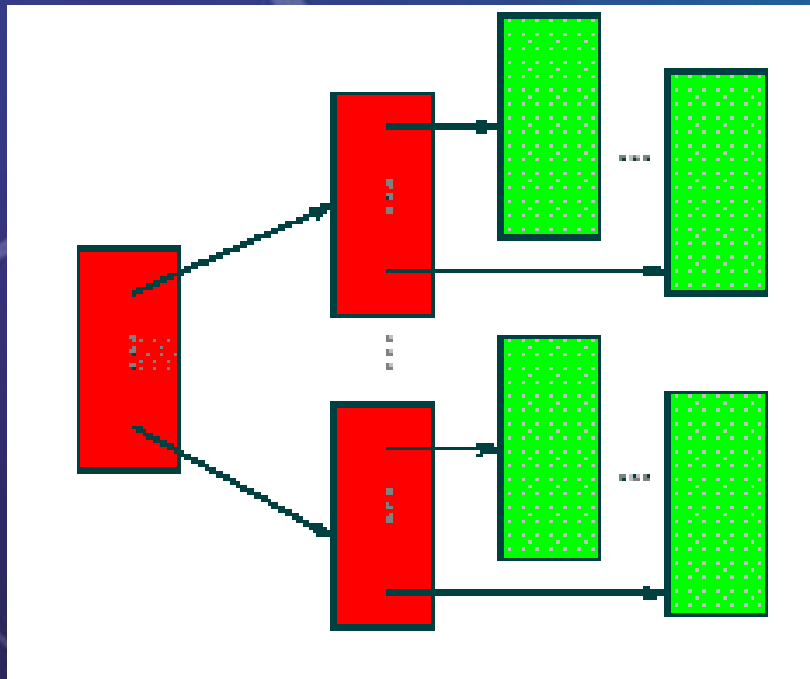
## Schemat listowy



 blok danych  
 blok indeksowy

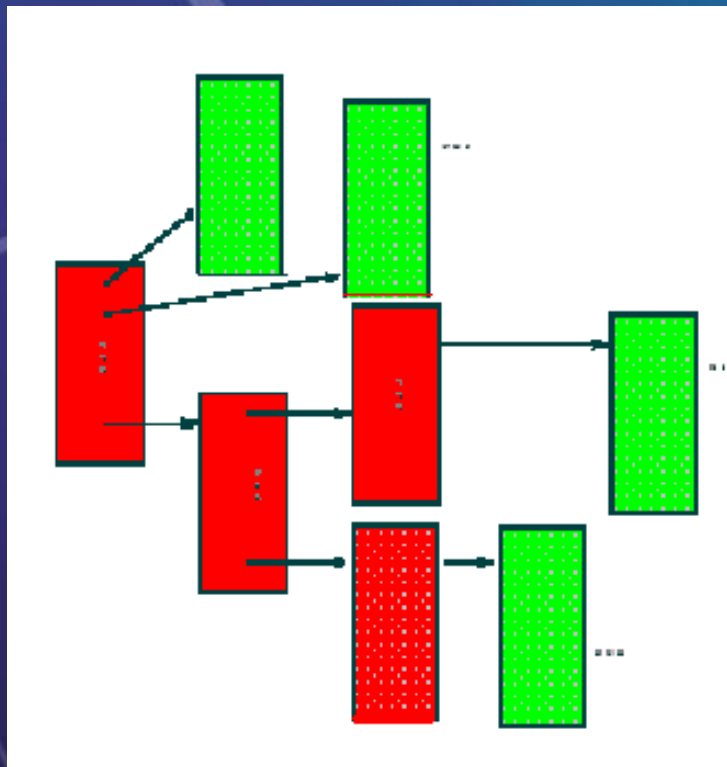
# Struktura bloku indeksowego

## indeks wielopoziomowy



# Struktura bloku indeksowego

## indeks kombinowany





# Właściwości - zalety i wady

- Wskaźniki dowiązań do każdego pliku są pamiętane w odrębnych blokach indeksów na dysku.
- Dla dużego pliku trzeba przeznaczyć kilka bloków indeksów.
  - schemat listowy
  - indeks wielopoziomowy
  - schemat kombinowany
- Brak fragmentacji zewnętrznej.
- Umożliwia dostęp bezpośredni.

# Wydajność

- Metody przydziału - różnice w zapotrzebowaniu na pamięć i czas dostępu do bloków danych.
- Przydział ciągły - pobranie danych wymaga jednego kontaktu z dyskiem (dostęp sekwencyjny i swobodny).
- Przydział listowy - (dostęp do  $i$ -tego bloku  $i$  operacji czytania z dysku -- dostęp sekwencyjny).
- Struktura pliku - zależna od deklarowanego typu dostępu.
- Konwersja typu pliku - kopiowanie do nowego pliku o wymaganym typie.

# Zarządzanie wolną przestrzenią

- Ponieważ obszar dysku jest ograniczony, więc w miarę możliwości należy dbać o wtórne zagospodarowanie dla nowych plików przestrzeni po plikach usuniętych.
- Lista wolnych obszarów może być implementowana w postaci:
  - wektora binarnego
  - listy powiązanej
  - grupowania
  - zliczania

# Zarządzanie wolną przestrzenią wektor bitowy

**wektor bitowy superbloku**

0000111110000011111001111111

|   |   |    |   |    |    |    |
|---|---|----|---|----|----|----|
| 1 | 2 | 3  | 4 |    |    |    |
|   |   | 10 |   | 12 |    | 14 |
|   |   |    |   |    | 20 | 21 |
|   |   |    |   |    |    |    |

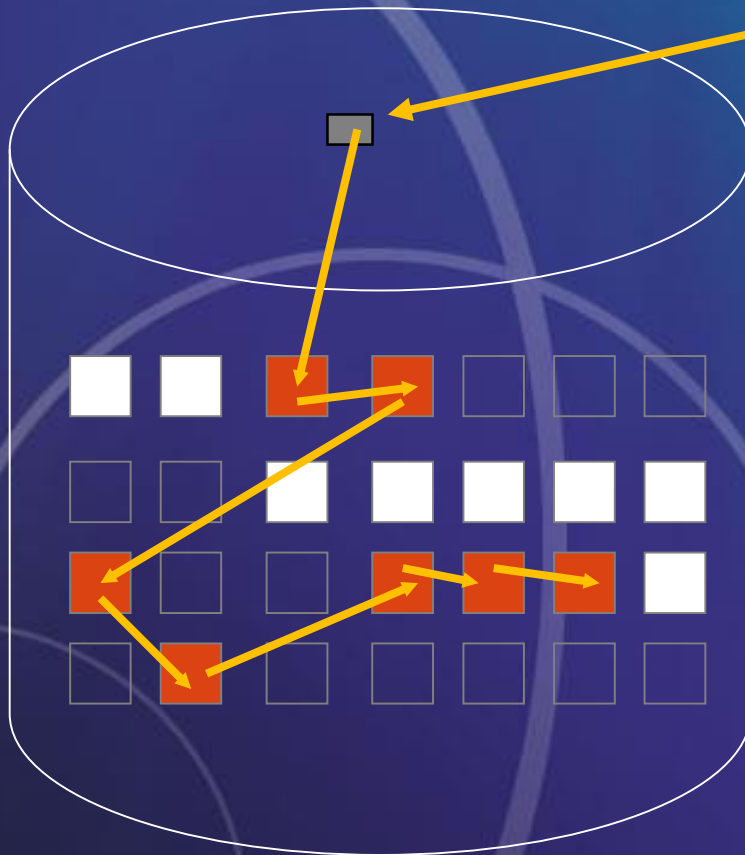
Każdy blok dyskowy jest reprezentowany przez jeden bit w wektorze.

Wartość 1 oznacza, że dany blok jest wolny, natomiast wartość 0 oznacza, że dany blok jest zajęty

- mało wydajne
- tylko dla małych dysków

# Zarządzanie wolną przestrzenią lista powiązana

indeks pierwszego wolnego  
bloku



Powiązanie wszystkich wolnych bloków w ten sposób, że w bloku poprzednim znajduje się indeks bloku następnego.

Indeks pierwszego bloku znajduje się w specjalnym miejscu w systemie plików.

metoda niewydajna - aby przejrzeć listę trzeba odczytać każdy blok

# Zarządzanie wolną przestrzenią grupowanie





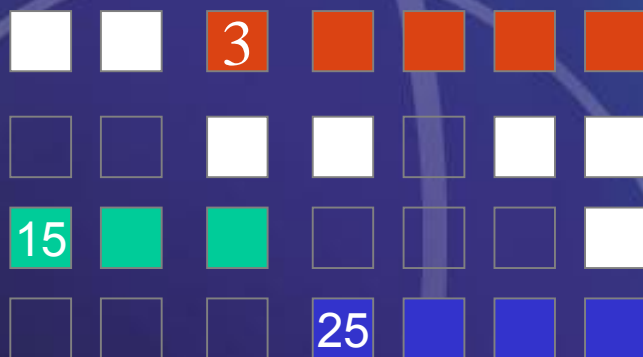
# Zarządzanie wolną przestrzenią zliczanie

wykaz wolnych  
obszarów

3,5

15,3

25,4



W przypadku kilku kolejnych (przylegających do siebie) wolnych bloków pamiętany jest tylko indeks pierwszego z nich oraz liczba wolnych bloków znajdujących się bezpośrednio za nim.

Wykaz wolnych obszarów jest ciągiem wpisów składających się z indeksu bloku oraz licznika



# Katalogi plików

Podstawowym zagadnieniem w przypadku dostępu do pliku jest odwzorowanie symbolicznej nazwy pliku na jego adres fizyczny w pamięci pomocniczej - służy do tego **katalog plików**.

# Katalog wykonywane operacje

- Odnajdywanie pliku
- Tworzenie pliku
- Usuwanie pliku
- Przemianowywanie pliku
- Wyprowadzanie zawartości katalogu
- Obchód systemu plików

# Struktury katalogowe

- Katalog jednopoziomowy.
- Katalog dwupoziomowy.
- Struktury drzewiaste.
- Acykliczne grafy katalogów.
- Graf ogólny katalogów.

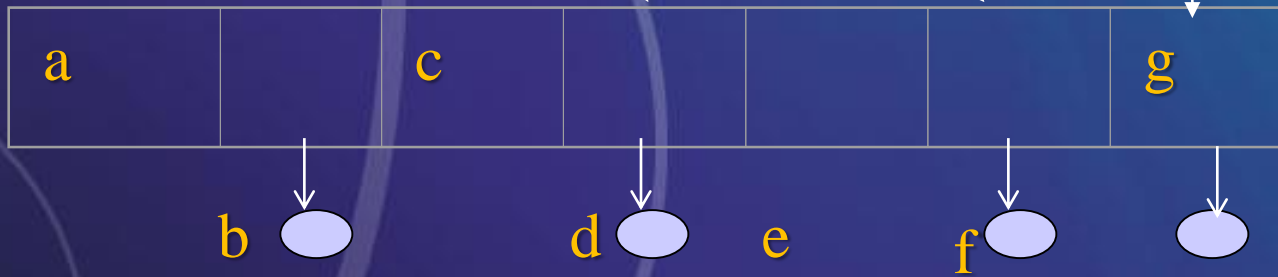
# Struktury katalogowe



pliki

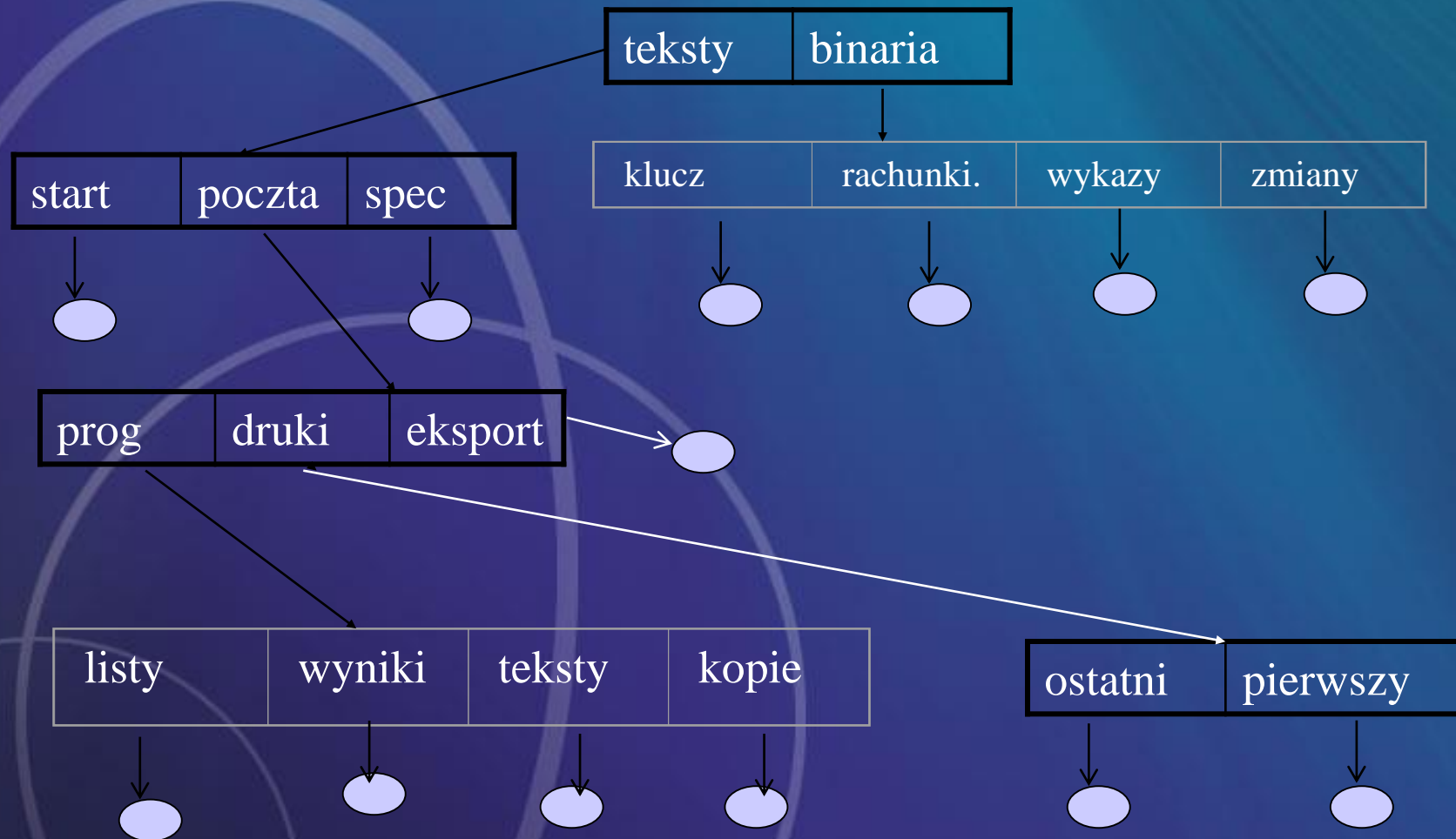


katalog jednopoziomowy

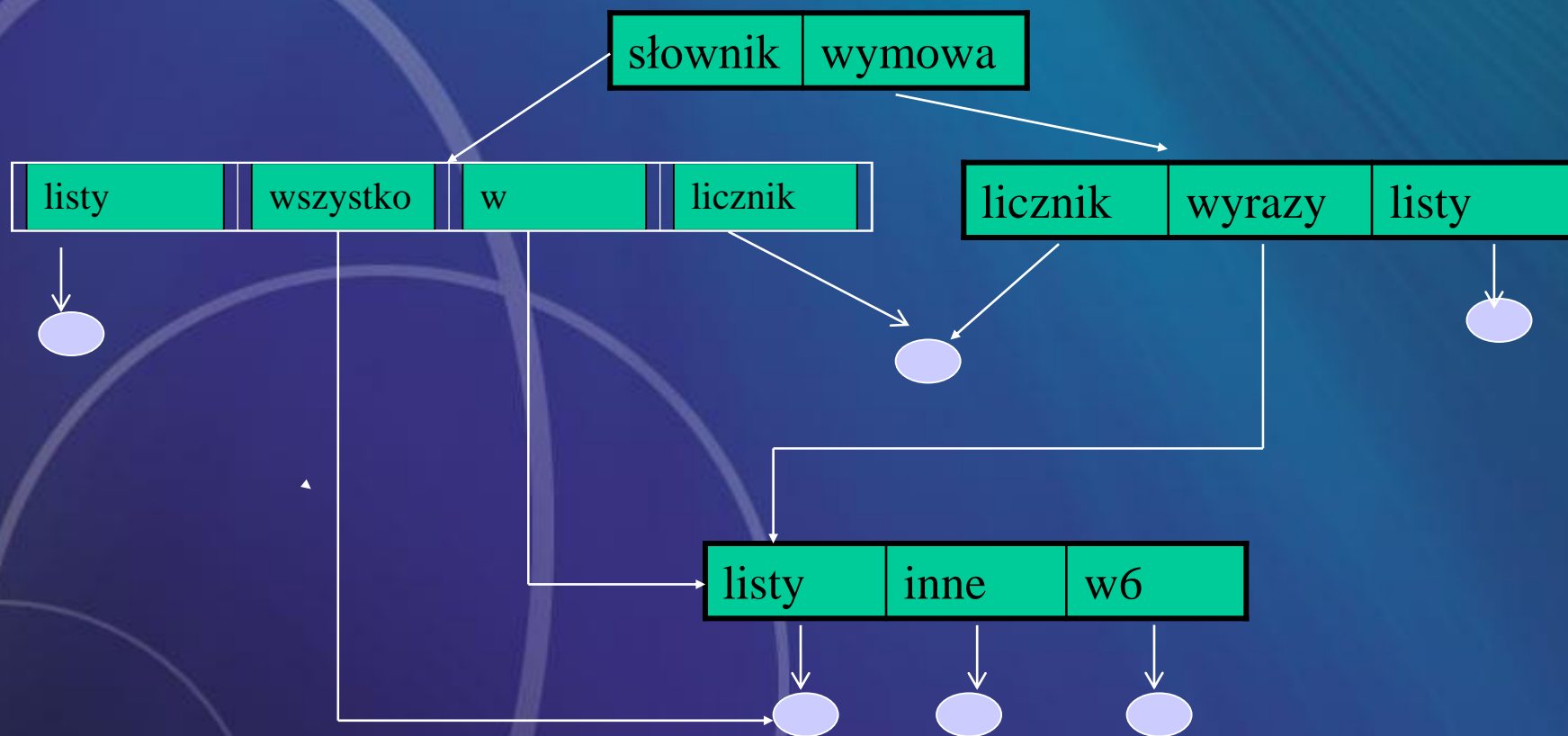


katalog dwupoziomowy

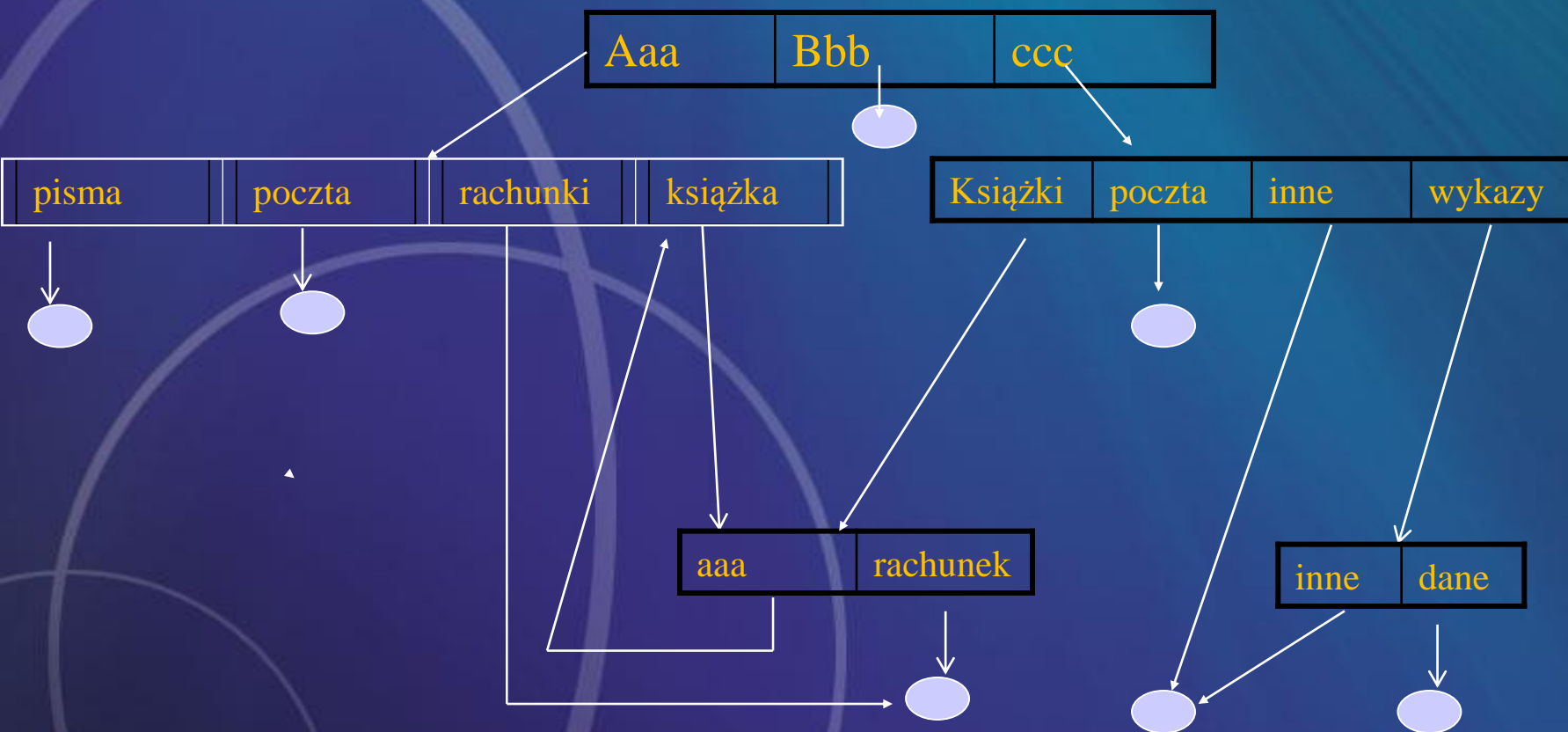
# Katalog o strukturze drzewiastej



# Acykliczne grafy katalogów



# Graf ogólny katalogów





# Implementacja katalogu

Katalog składa się z ciągu wpisów katalogowych w ogólnej postaci:

**nazwa pliku**

**inne atrybuty**

- Lokalizacja wpisu katalogowego polega na poszukiwaniu liniowym (sprawdzane są kolejne pozycje zaczynając od pierwszej).
- Poszukiwanie wpisu można przyspieszyć poprzez posortowanie wg nazwy.

# Implementacja katalogu – tablica haszowana

- Wpisy ułożone są na pozycjach odpowiadających wartościom funkcji haszującej.
- Funkcja haszująca odwzorowuje nazwę pliku na wartość z określonego przedziału, traktowaną jako indeks wpisu.
- Ta sama funkcja haszująca wykorzystywana jest do lokalizacji wpisu.
- Przy usuwaniu konfliktów w katalogu mogą być potrzebne dodatkowe struktury.

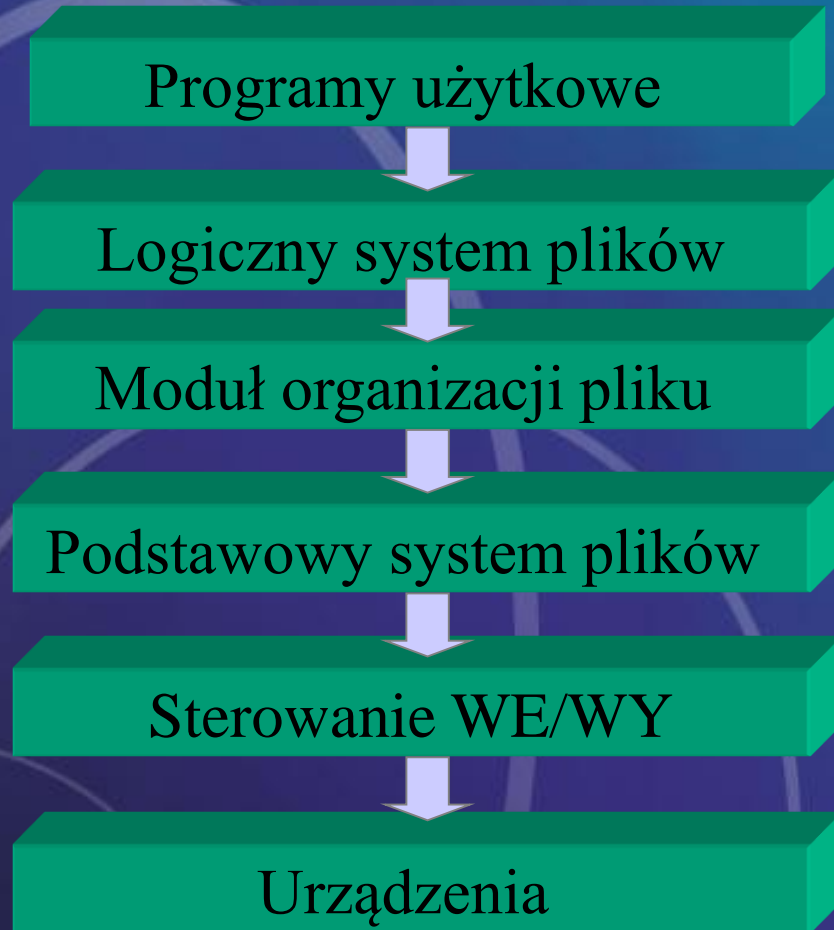
# Implementacja katalogu – struktura indeksowa

- Wpisy katalogowe powiązane są w strukturę drzewiastą przyspieszającą wyszukiwanie (np. drzewo binarne, B-drzewo).
- Lokalizacja wpisu polega na przejściu drzewa zgodnie z zasadami jego budowy.
- Struktura drzewa jest zoptymalizowana w taki sposób, żeby minimalizować liczbę operacji dyskowych podczas przeszukiwania.

# Semantyka spójności

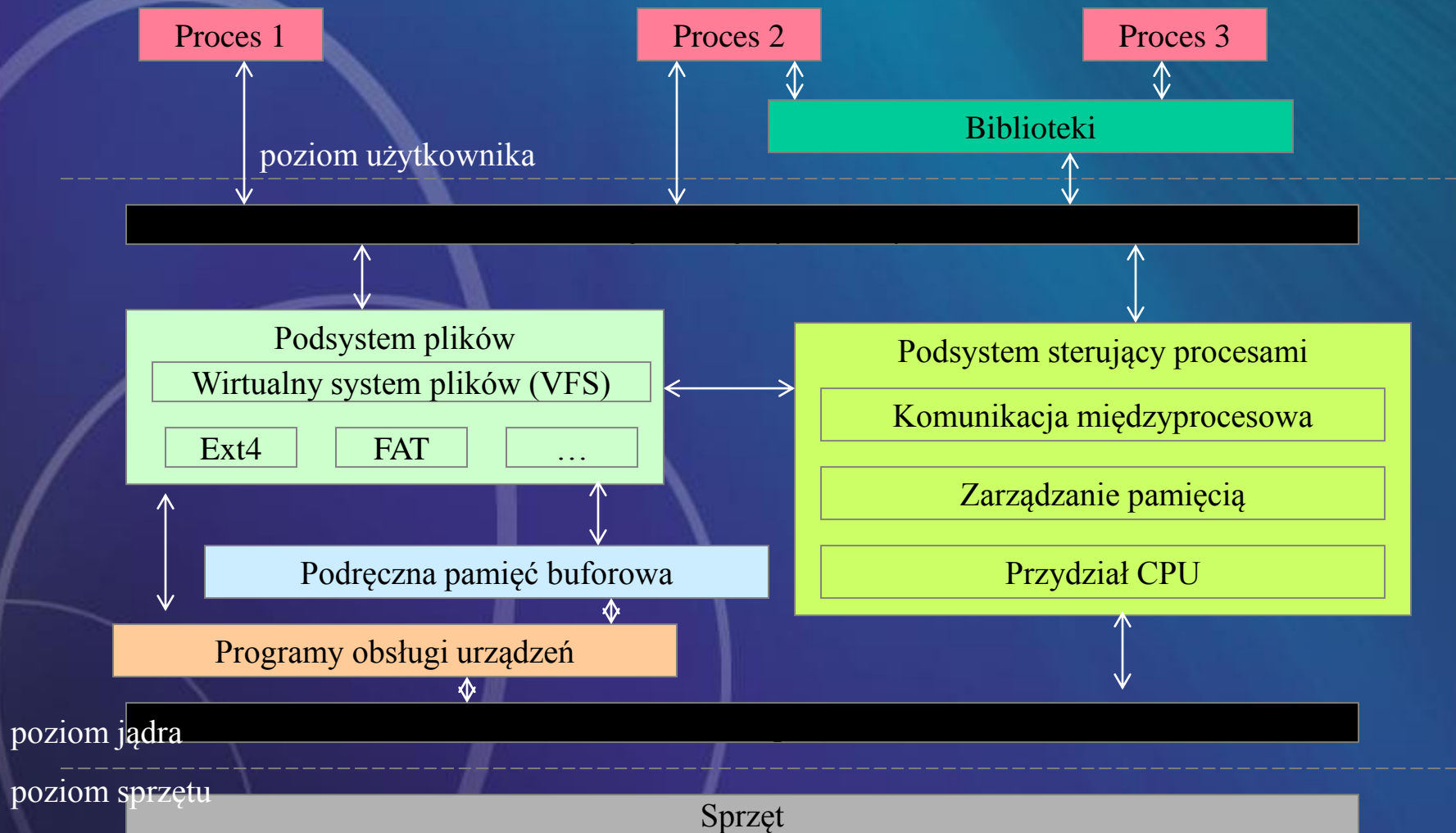
- Ważne kryterium oceny dowolnego systemu plików, który realizuje dzielenie plików.
- Określa warunki, przy których zmiany danych wykonywane przez jednego użytkownika są obserwowalne przez innych użytkowników.

# Warstwowy system plików



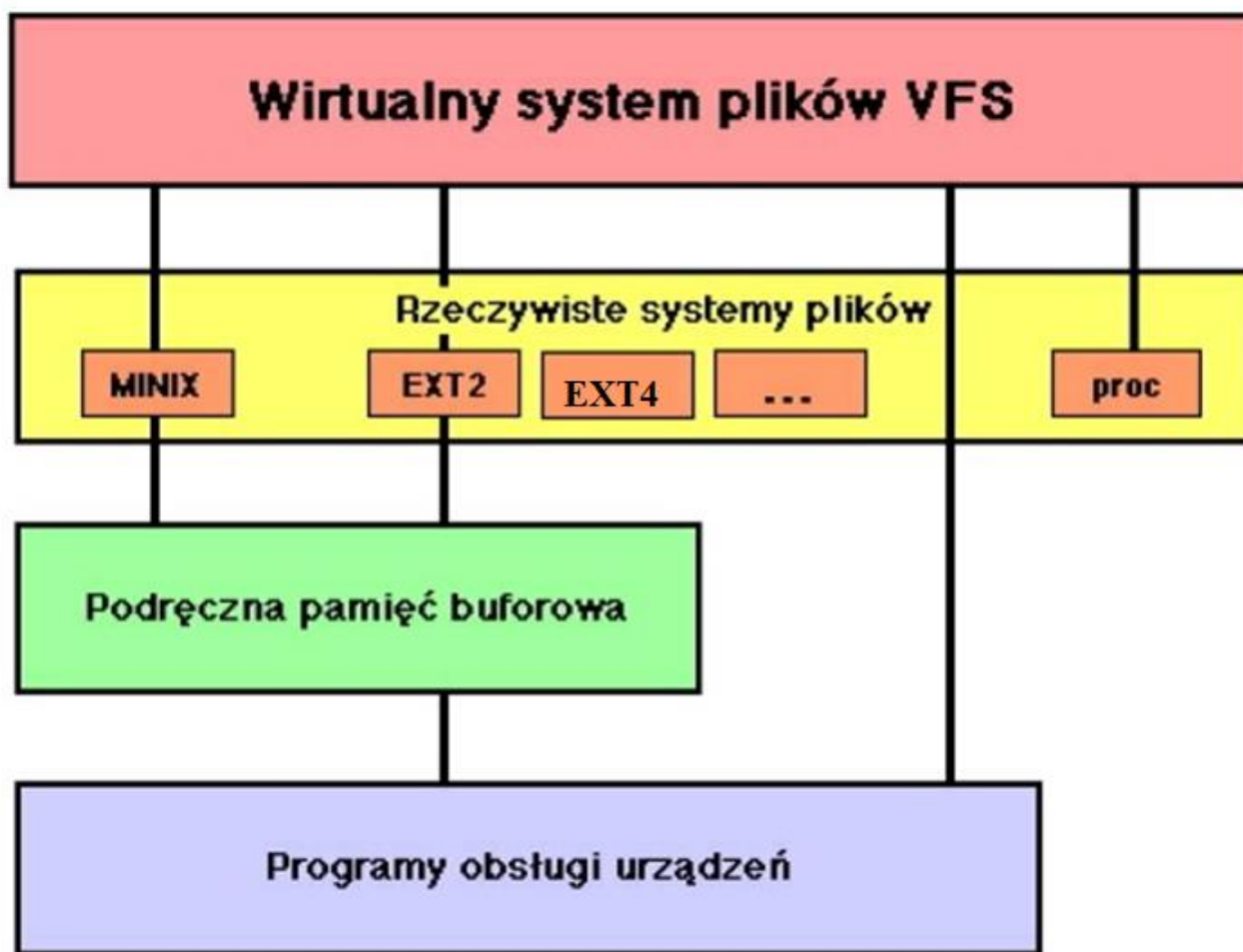
- System plików składa się na ogół z wielu poziomów, warstw.
- Każdy poziom korzysta z właściwości niższych poziomów do wytworzenia nowych właściwości, używanych przez poziomy wyższe.

# System Linux





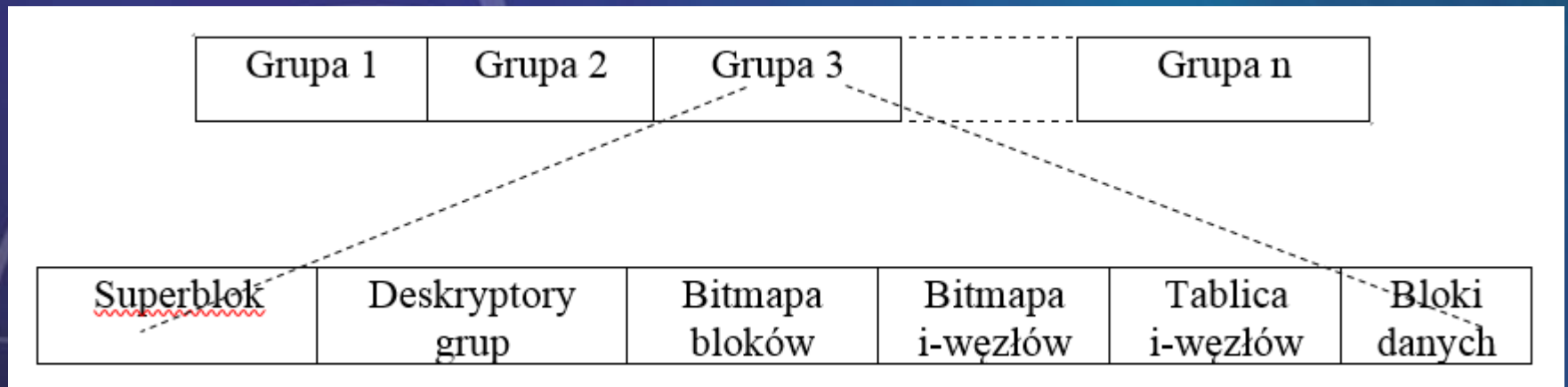
# Struktura systemu plików





# Struktura systemu plików

- System plików jest podzielony na grupy.
- Każda grupa ma określoną stałą wielkość za wyjątkiem ostatniej.



# Systemy plików Linuxa

|                                 | <b>NTFS</b>      | <b>Ext2</b> | <b>Ext3</b> | <b>ReiserFS</b> | <b>XFS</b> | <b>JFS</b>   |
|---------------------------------|------------------|-------------|-------------|-----------------|------------|--------------|
| Maksymalny rozmiar zbioru       | 16 EB            | 16 GB-2 TB* | 16 GB-2 TB* | 1 EB            | 9 EB       | 8 EB         |
| Maksymalny rozmiar partycji     | 16 EB            | 2 TB-31 TB* | 2 TB-31 TB* | 16 TB           | 9 EB       | 512 TB-4 PB* |
| Posiada księgowanie             | tak              | nie         | tak         | tak             | tak        | tak          |
| system operacyjny               | Windows NT /Unix | Unix        | Unix        | Unix            | Unix       | Unix         |
| zapewnia mechanizm praw dostępu | tak              | tak         | tak         | tak             | tak        | tak          |
| wrażliwy na wielkość liter      | nie              | tak         | tak         | tak             | tak        | tak          |

# Format patrycji - UNIX



Struktura wpisu katalogowego



# Struktura i-węzła

- identyfikator właściciela oraz grupy pliku
- typ pliku
- prawa dostępu
- rozmiar pliku w bajtach ostatni czas dostępu, modyfikacji
- czas utworzenia i skasowania
- liczba dowiązań
- liczba bloków dyskowych zajmowanych przez plik
- adresy dyskowe

# Adresowanie bloków dyskowych

