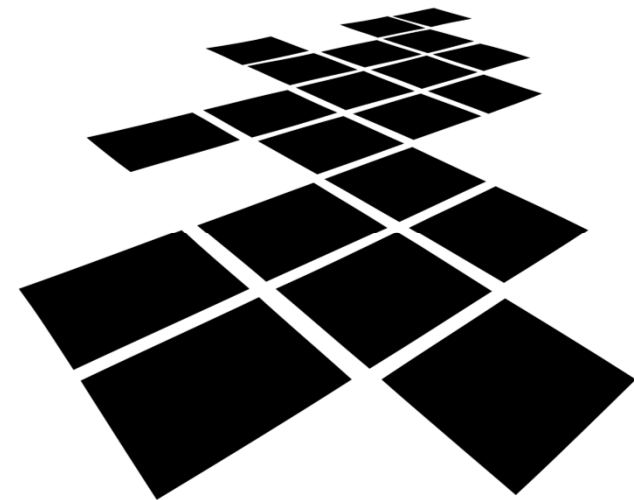


System plików — warstwa fizyczna

**Wykład prowadzą:
Jerzy Brzeziński
Dariusz Wawrzyniak**



**UCZELNIA
ONLINE**



Plan wykładu

- Przydział miejsca na dysku
- Zarządzanie wolną przestrzenią
- Implementacja katalogu
- Przechowywanie podręczne
- Integralność systemu plików
- Semantyka spójności



Organizacja fizyczna systemu plików

- Przydział miejsca na dysku
 - przydział ciągły, przydział listowy, przydział indeksowy
- Zarządzanie wolną przestrzenią
 - wektor bitowy, lista powiązana, grupowanie, zliczanie
- Implementacja katalogu
 - lista liniowa, tablica haszowa, struktura indeksowa

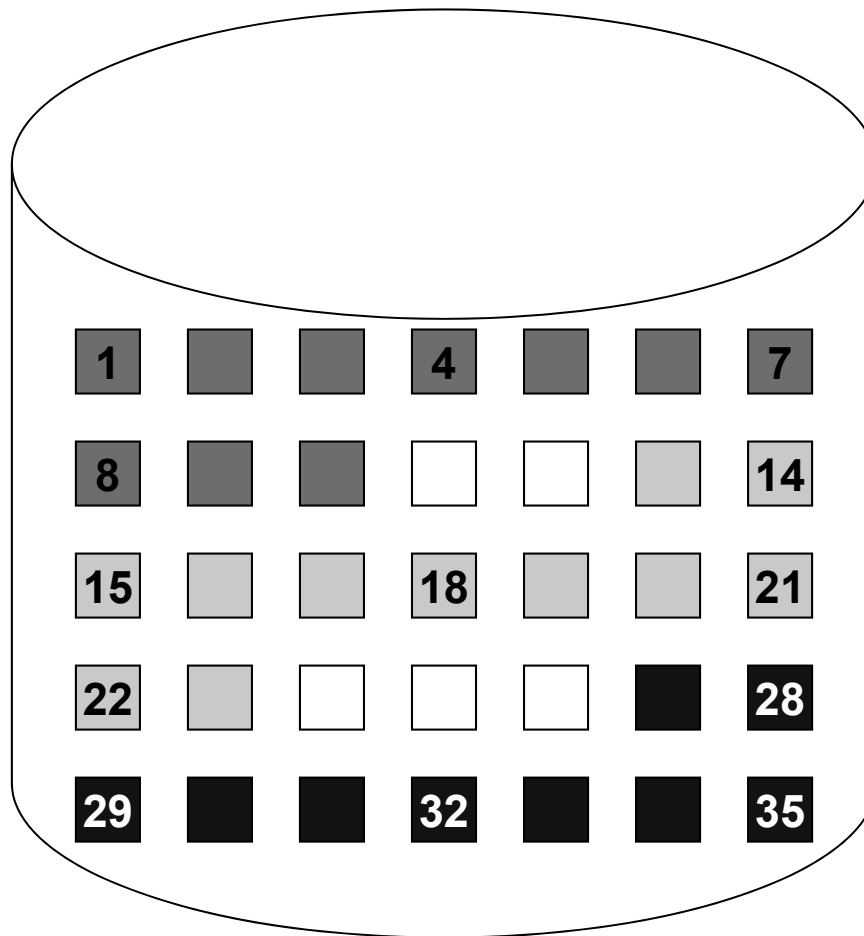


Przydział miejsca na dysku

- Przydział ciągły (ang. contiguous allocation) — cały plik zajmuje ciąg kolejnych bloków
- Przydział listowy (łańcuchowy, ang. linked allocation, chained allocation) — bloki pliku tworzą listę powiązaną
- Przydział indeksowy (ang. indexed allocation) — bloki z danymi wskazywane są przez bloki indeksowe, które mogą być zorganizowane w:
 - schemat listowy
 - schemat wielopoziomowy
 - schemat kombinowany



Przydział ciągły



Katalog:

blok początkowy: 1
rozmiar: 10 bloków

blok początkowy: 13
rozmiar: 11 bloków

blok początkowy: 27
rozmiar: 9 bloków

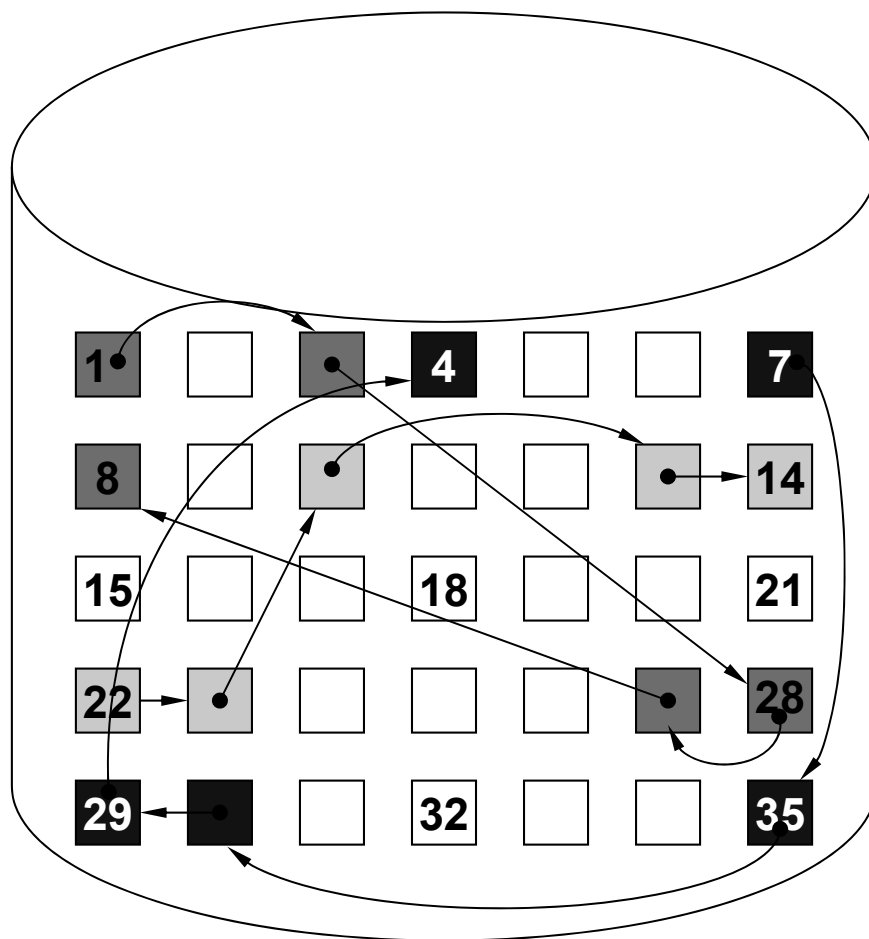


Przydział ciągły — właściwości

- Efektywność dostępu (niewielkie ruchy głowic dysk.)
- Łatwa lokalizacja bloków pliku zarówno przy dostępie sekwencyjnym jak i bezpośrednim
- Problem fragmentacji zewnętrznej — po usuniętych plikach pozostają dziury, które trudno połączyć w jeden większy blok
- Problem rozszerzania pliku
 - pliku nie da się rozszerzyć,
 - będzie go trzeba przenieść w nowe miejsce (znaleźć większą dziurę)
 - będzie trzeba z góry zarezerwować więcej miejsca w przestrzeni dyskowej



Przydział listowy (łańcuchowy)



Katalog:

blok początkowy: 1
blok końcowy: 8

blok początkowy: 22
blok końcowy: 14

blok początkowy: 7
blok końcowy: 4



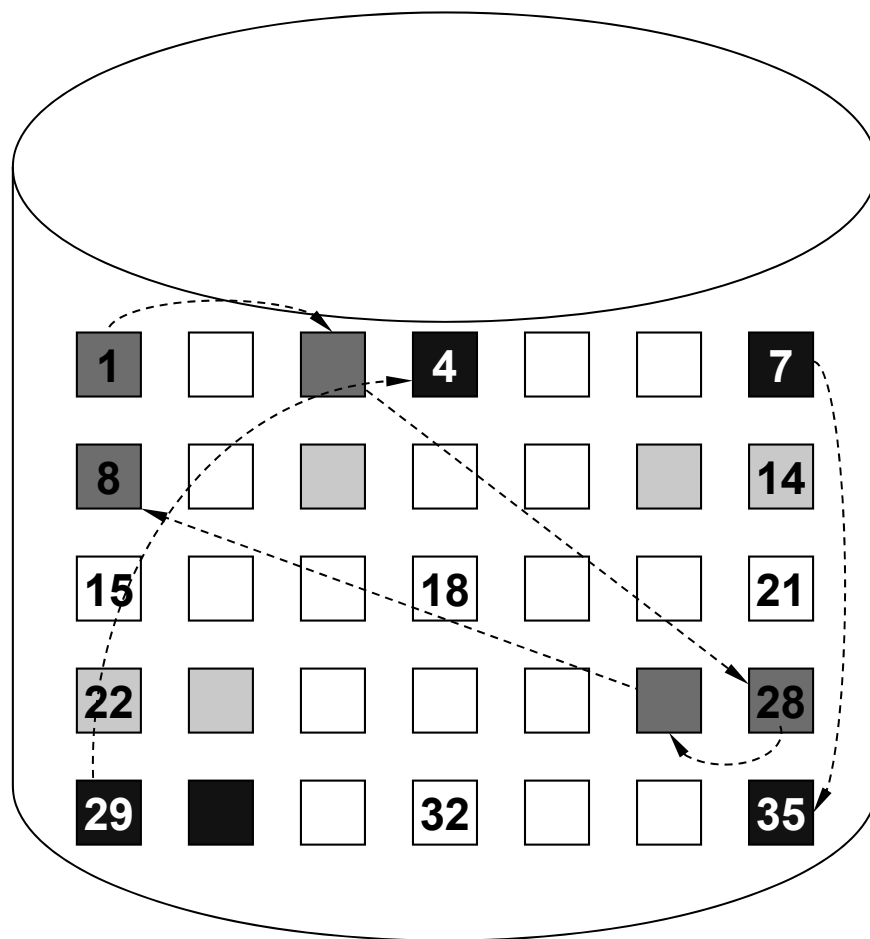
Przydział listowy — właściwości

- Nie ma problemu fragmentacji zewnętrznej
- Łatwa realizacja dostępu sekwencyjnego
- Problem realizacji dostępu bezpośredniego
- Konieczność pamiętania wewnątrz bloku wskaźnika do bloku następnego
- Zawodność — utrata jednego bloku pociąga za sobą stratę wszystkich następnych



Tablica alokacji plików — FAT (File Allocation Table)

- FAT jest dodatkową strukturą (tablicą) umieszczoną w odpowiednim obszarze na dysku
- Każdy element tablicy FAT odpowiada dokładnie jednej jednostce alokacji (blokowi) z przestrzeni bloków plikowych i indeksowany jest numerem bloku
- Element tablicy FAT zawiera indeks następnego bloku przydzielonego danemu plikowi lub pewną wartość specjalną oznaczającą wolną pozycję lub ostatnią pozycję danego pliku



Struktura tablicy alokacji plików

1	3
2	
3	28
4	#
5	
6	
7	35
8	#
...	
27	8
28	27

Katalog:

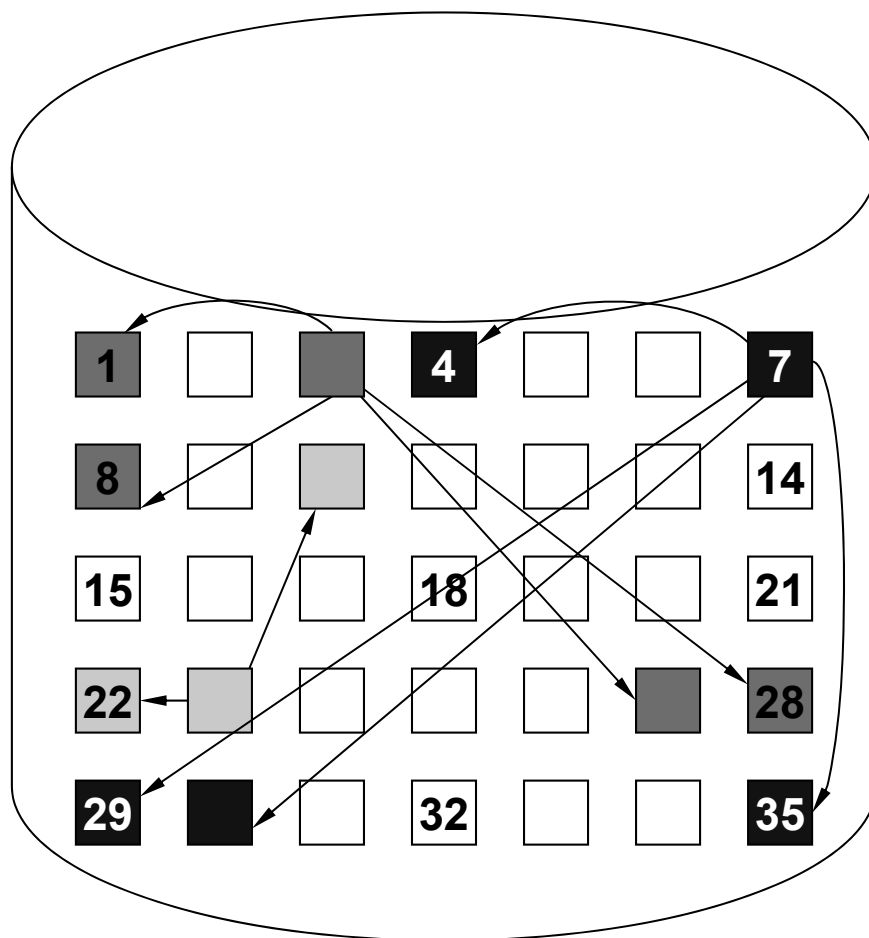
blok początkowy: 1
blok końcowy: 8

blok początkowy: 22
blok końcowy: 14

blok początkowy: 7
blok końcowy: 4



Przydział indeksowy



Katalog:

blok indeksowy: 3
rozmiar: 4 bloki

blok indeksowy: 23
rozmiar: 2 bloki

blok indeksowy: 7
rozmiar: 4 bloki

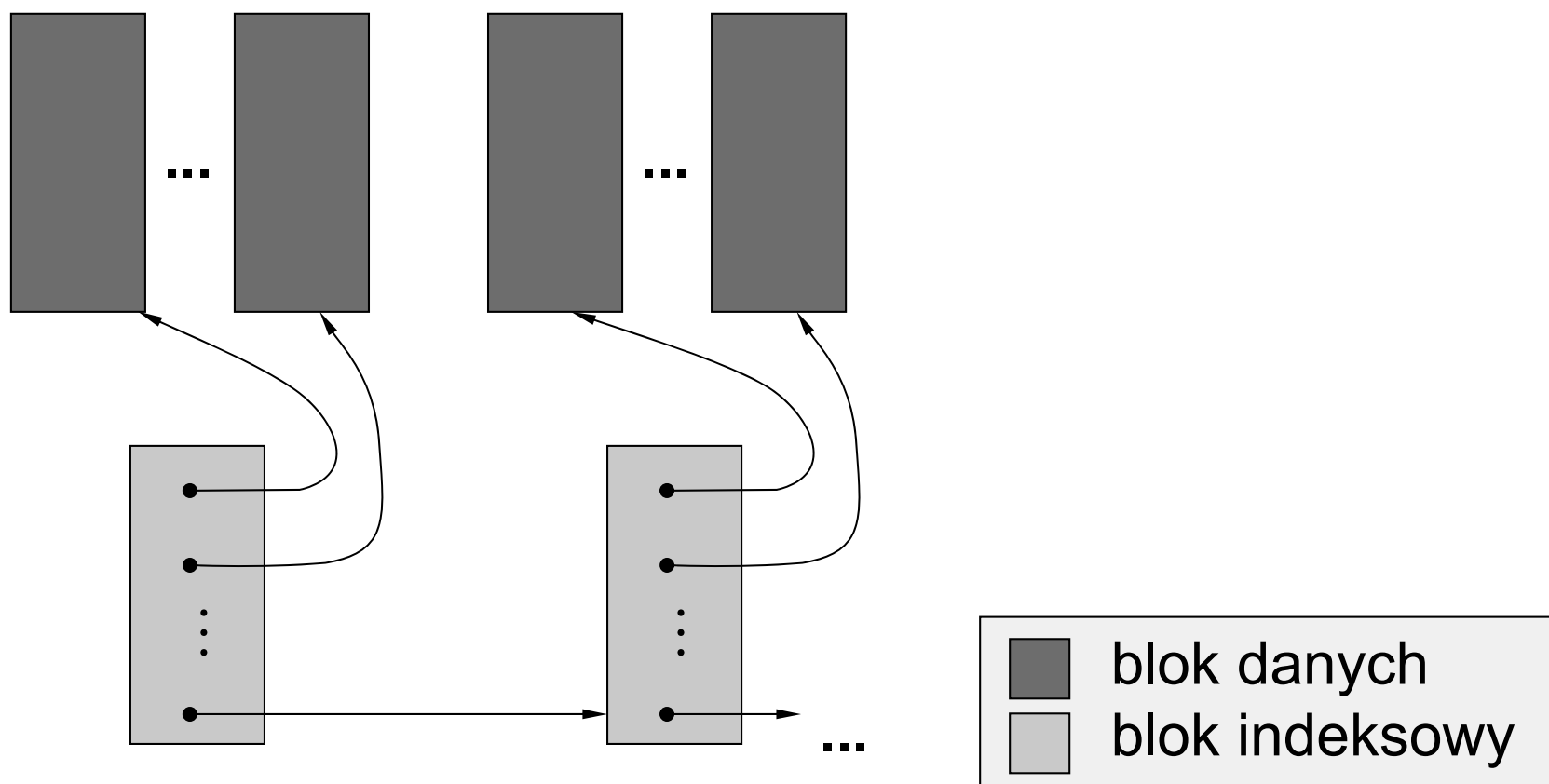


Struktura bloku indeksowego

- Schemat listowy — w ostatnim elemencie bloku indeksowego znajduje się wskaźnik do następnego bloku indeksowego tego pliku.
- Schemat wielopoziomowy — blok indeksowy pierwszego poziomu zawiera wskaźnik do bloków drugiego poziomu itd.
- Schemat kombinowany — zastosowanie do pewnej liczby bloków indeksu pierwszego poziomu, dla następnych bloków indeksu dwupoziomowego itp.

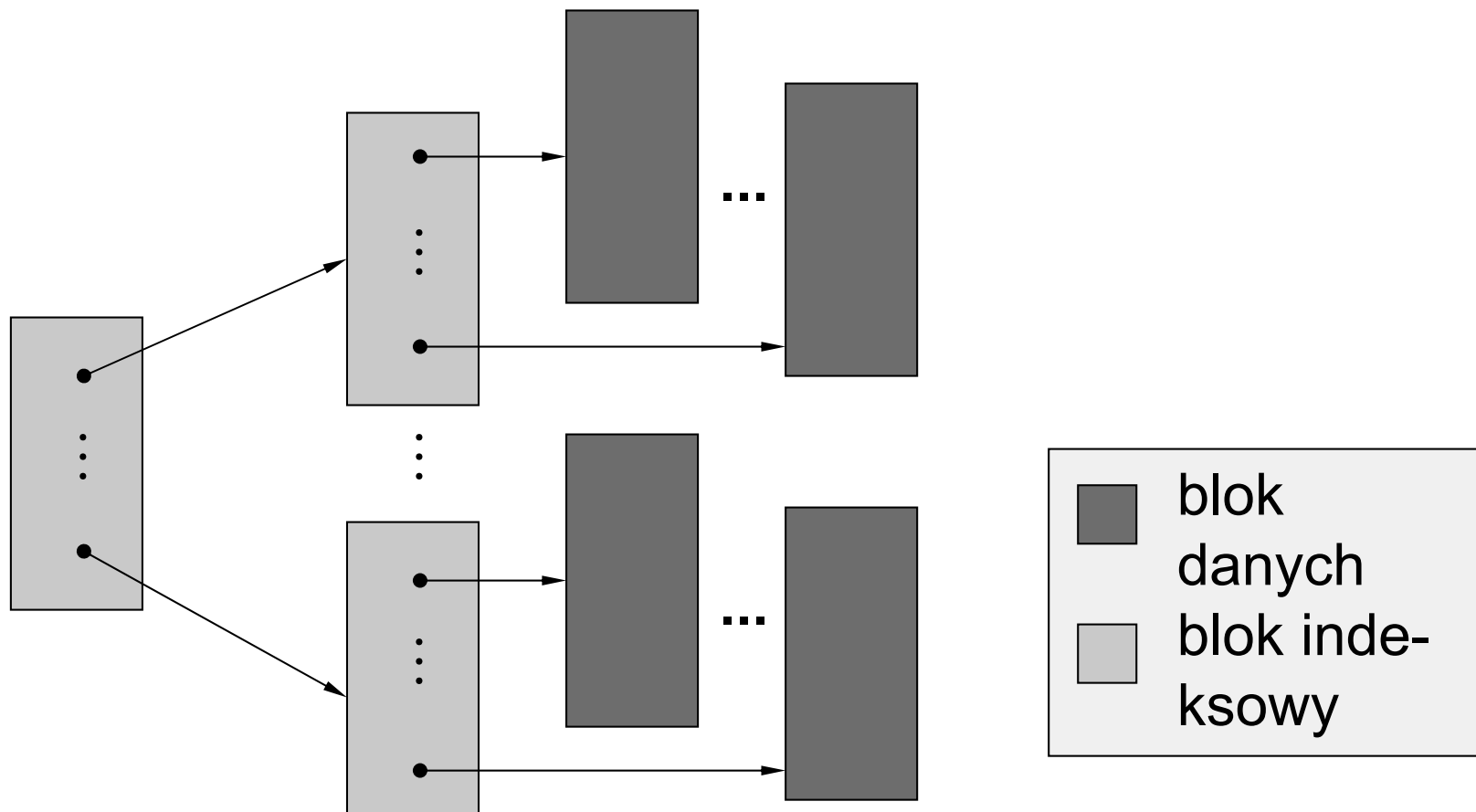


Struktura bloku indeksowego — schemat listowy



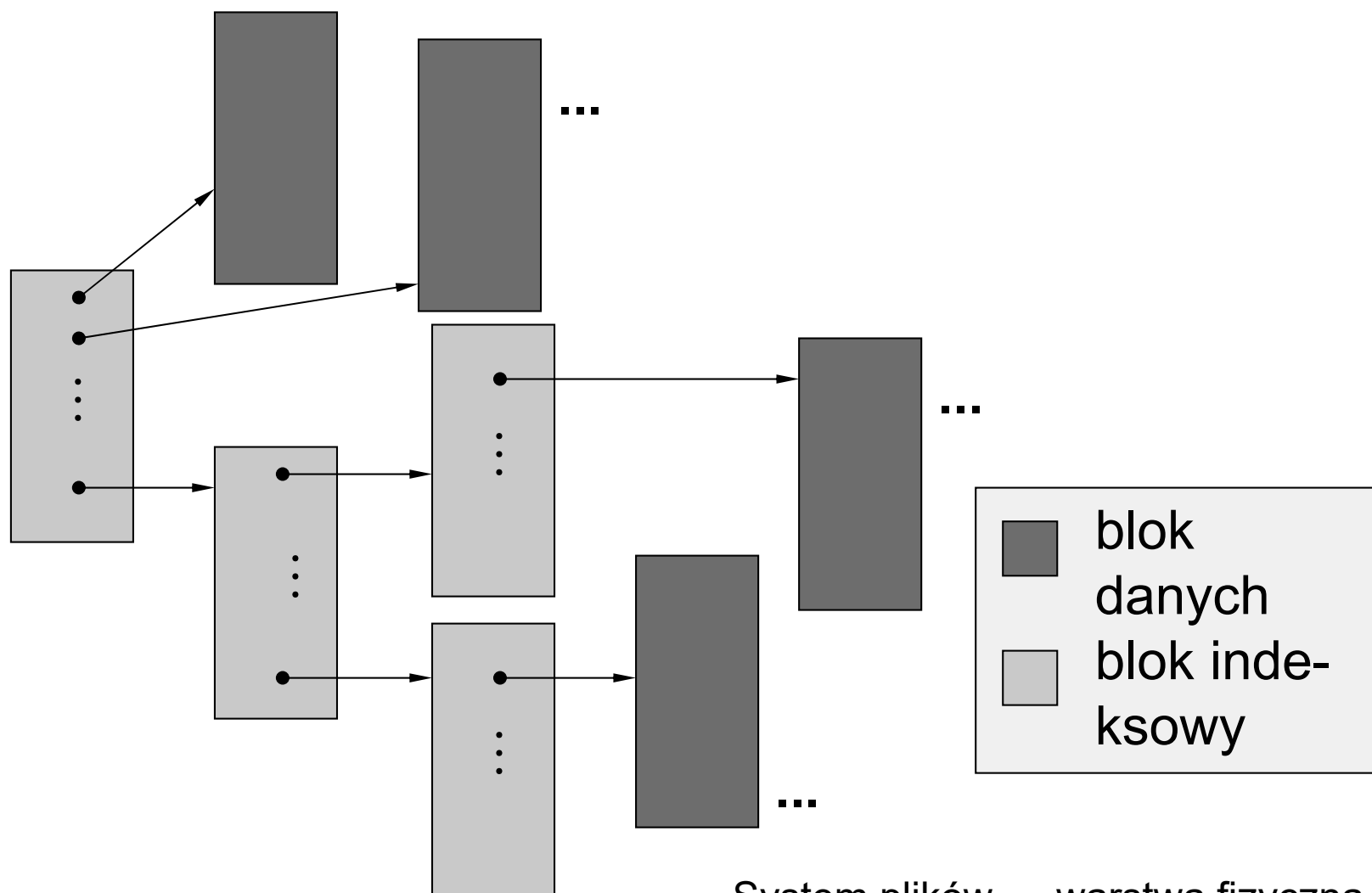


Struktura bloku indeksowego — indeks wielopoziomowy





Struktura bloku indeksowego — schemat kombinowany





Przydział indeksowy — właściwości

- Stosunkowo szybka lokalizacja dowolnego bloku pliku
- Łatwa realizacja dostępu bezpośredniego
- Brak problemu fragmentacji zewnętrznej
- Konieczność przeznaczenie pewnej części przestrzeni dyskowej na bloki indeksowe
- Zawodność: utrata bloku indeksowego oznacza utratę sporej części pliku lub nawet całej zawartości.

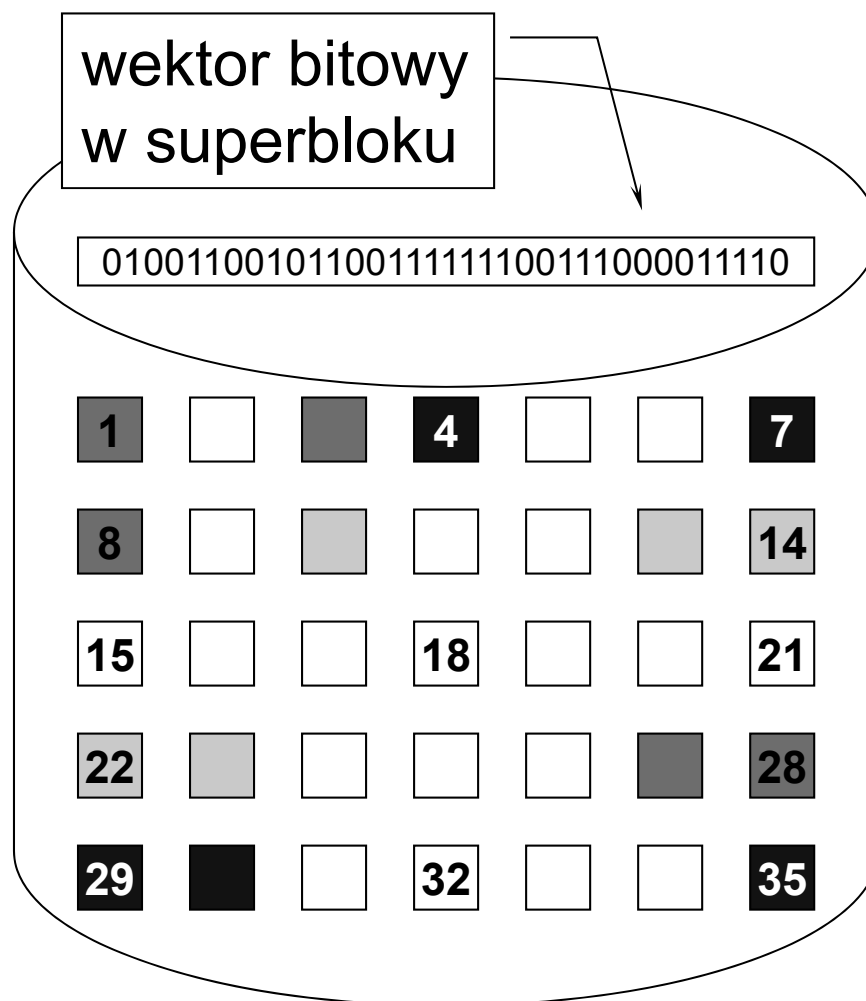


Zarządzanie wolną przestrzenią

- Wektor bitowy — każdy bit odpowiada jednemu blokowi, wartość 1 oznacza wolny blok.
- Lista powiązana — każdy wolny blok zawiera indeks następnego wolnego bloku.
- Grupowanie — niektóre wolne bloki wypełnione są w całości indeksami innych wolnych bloków, ostatni indeks wskazuje na kolejny blok wypełniony w całości indeksami.
- Zliczanie — wykaz wolnych bloków obejmuje indeks pierwszego wolnego bloku oraz liczbę wolnych bloków znajdujących się za nim, stanowiących ciągły obszar.

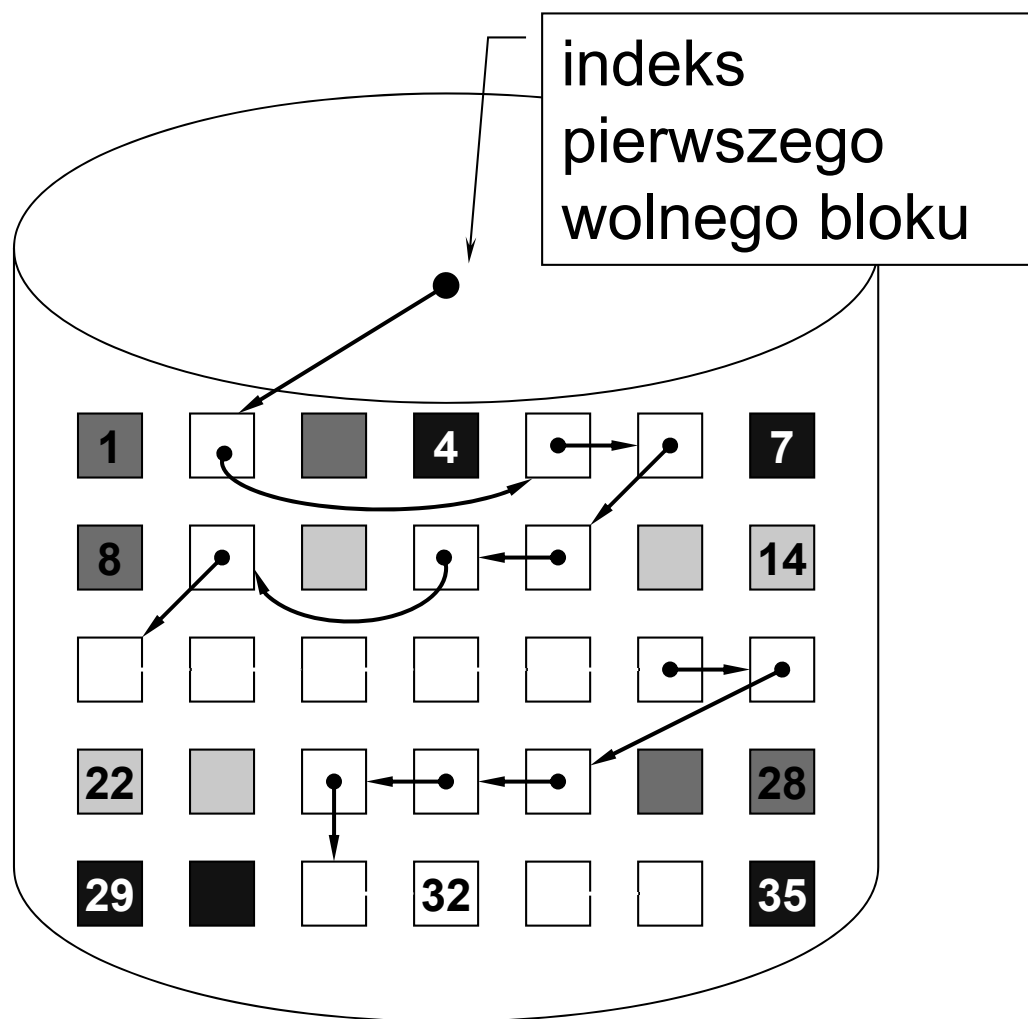


Zarządzanie wolną przestrzenią — wektor bitowy



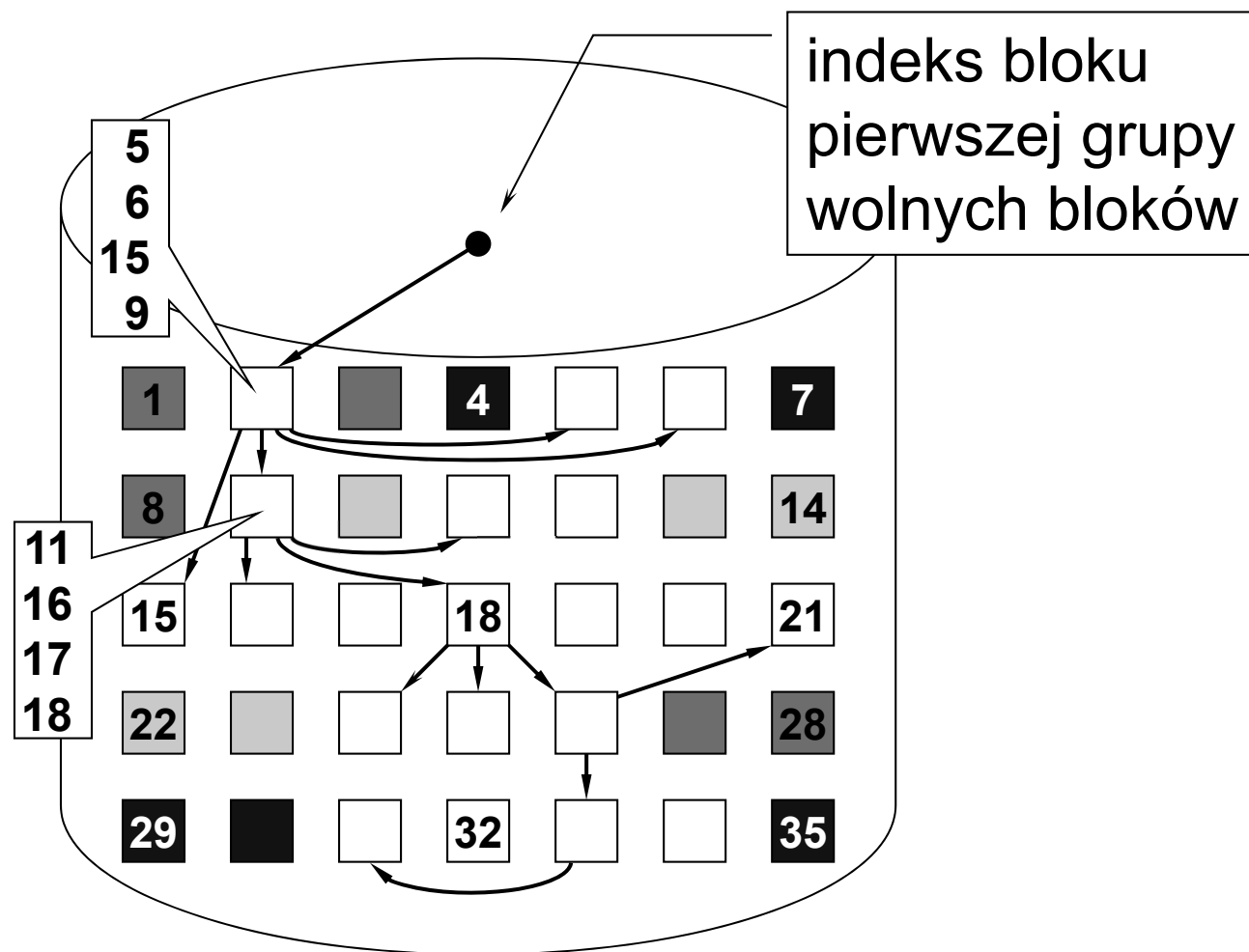


Zarządzanie wolną przestrzenią — lista powiązana



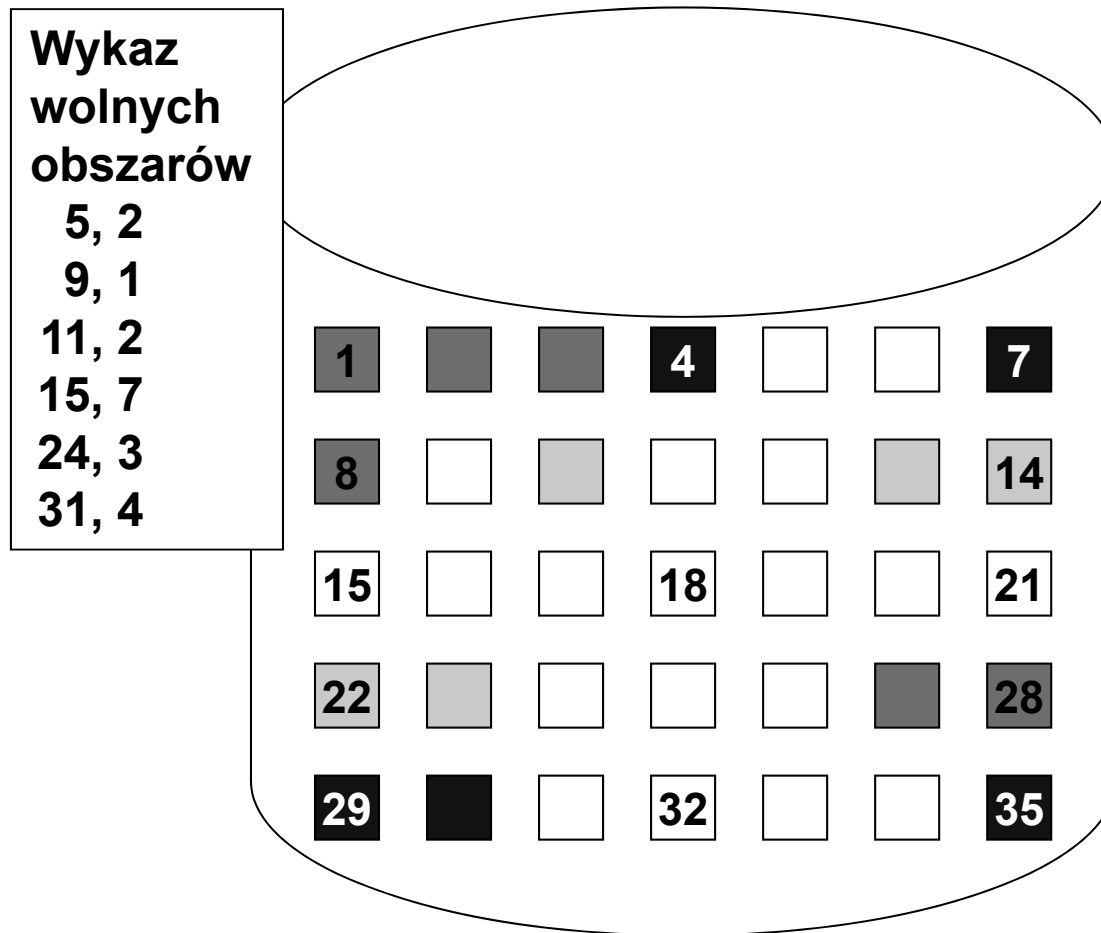


Zarządzanie wolną przestrzenią — grupowanie





Zarządzanie wolną przestrzenią — zliczanie





Implementacja katalogu — lista liniowa

- Katalog składa się z ciągu wpisów katalogowych ogólnej postaci:

nazwa pliku	inne atrybuty
-------------	---------------

- Lokalizacja wpisu polega na przeszukiwaniu liniowym (sprawdzane są kolejne pozycje, począwszy od pierwszej)
- Lokalizację wpisu można przyspieszyć poprzez posortowanie wg. nazwy, jednak utrzymanie takiej struktury jest kosztowne.



Implementacja katalogu — tablica haszowa

- 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,
- W katalogu mogą być potrzebne dodatkowe struktury pomocne przy usuwaniu konfliktów.

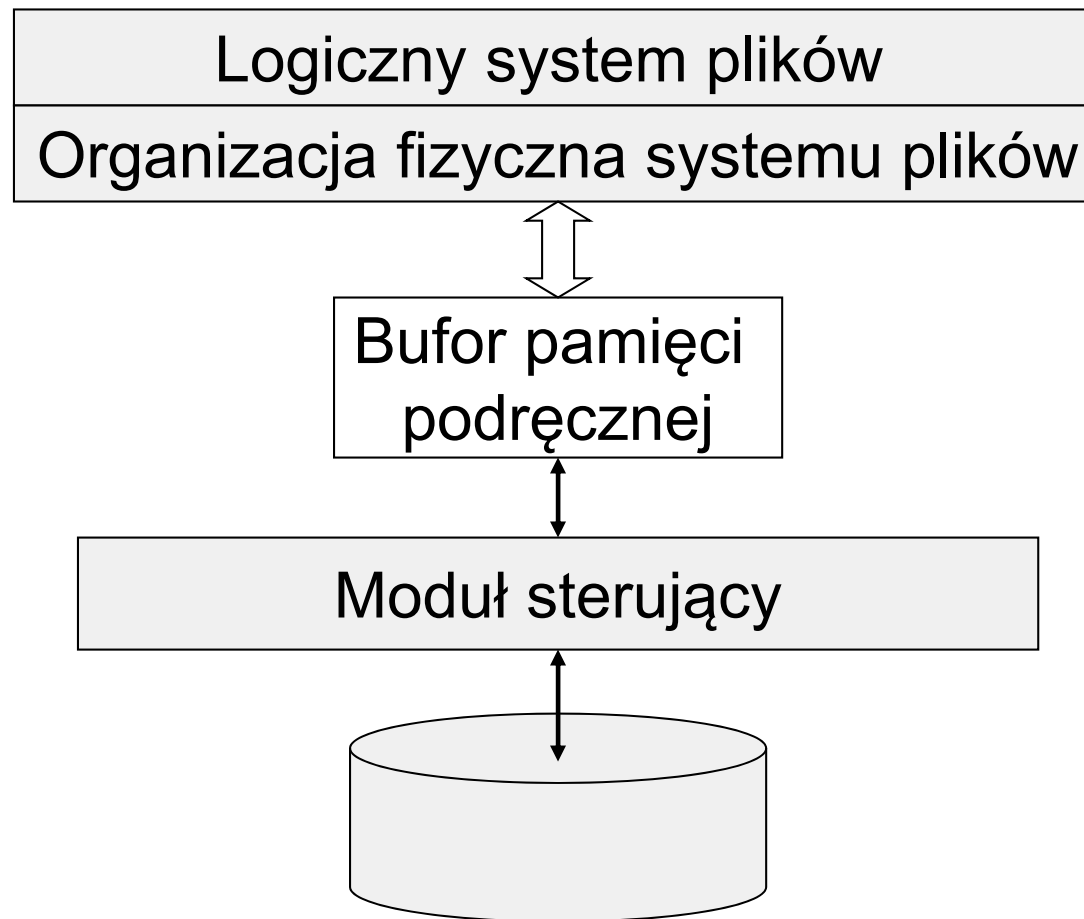


Implementacja katalogu — struktura indeksowa

- Wpisy katalogowe powiązane są w strukturę drzewiastą przyspieszającą wyszukiwanie (np. drzewo binarne, B-drzewo, 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.



Przechowywanie podręczne w systemie plików





Zasady przechowywania podręcznego

- Zawartość aktualnie wykorzystywanych bloków dyskowych utrzymywana jest w podręcznej pamięci buforowej.
- Obsługa żądania odczytu bloku polega najpierw na sprawdzeniu, czy dany blok znajduje się w podręcznej pamięci buforowej, a później ewentualnie sprowadzeniu z dysku.
- Żądany fragment kopiowany jest z podręcznej pamięci buforowej w odpowiednie miejsce w przestrzeni adresowej procesu.
- Obsługa żądania zapisu oznacza transfer danych do podręcznej pamięci buforowej.



Przechowywanie podręczne w realizacji operacji odczytu

1. Znajdź adres bloku dyskowego, zawierającego fragment pliku, którego odczytu zażądano.
2. Przekopiuj zawartość tego bloku do bufora w pamięci podręcznej systemu plików (jeśli ten blok się tam jeszcze nie znajduje).
3. Przekopiuj żądany fragment z bufora do przestrzeni adresowej procesu.



Przechowywanie podręczne w realizacji operacji zapisu

1. Znajdź adres bloku dyskowego, zawierającego fragment pliku, którego zapisu zażądano.
2. Przekopiuj zawartość tego bloku do bufora w pamięci podręcznej systemu plików (jeśli ten blok się tam jeszcze nie znajduje).
3. Przekopiuj żądany fragment z przestrzeni adresowej procesu do bufora.
4. Zapisz na dysk uaktualniony blok z bufora (albo w trybie natychmiastowym albo z opóźnieniem)



Integralność systemu plików

- W wyniku awarii systemu zawartość podręcznej pamięci buforowej może nie zostać zapisana na dysku lub może zostać zapisana tylko częściowo.
- Skutkiem w/w awarii może być pozostawienie systemu plików w stanie niespójnym.
- Większość systemów operacyjnych dostarcza odpowiednie narzędzie do sprawdzania integralności systemu plików, uruchamiane w ramach restartu systemu po awarii.



Przejawy braku integralności systemu plików

- Brak bloku zarówno w wykazie bloków zaalokowanych jak i bloków wolnych
- Obecność bloku zarówno w wykazie bloków zaalokowanych jak i bloków wolnych
- Wielokrotne powtórzenie się bloku w wykazie bloków wolnych (duplikacja wolnego bloku)
- Wielokrotne powtórzenie się bloku w wykazie bloków zaalokowanych (duplikacja zaalokowanego bloku)
- Niespójność informacji we wpisach katalogowych (np. niezgodność licznika dowiązań w systemie UNIX).



Semantyka spójności

- Semantyka spójności określa sposób postrzegania zmian zawartości pliku, dokonywanych przez współbieżnie działające procesy.
- Przykłady semantyki spójności:
 - semantyka spójności systemu UNIX — wynik operacji zapisu jest natychmiast widoczny dla innych procesów,
 - semantyka sesji — zmiany w pliku stają się widoczne tylko dla procesów, otwierających ten plik po zamknięciu sesji, w której był zapis,
 - semantyka stałych plików dzielonych — współdzielony plik może być tylko czytany.



Synchronizacja dostępu do plików

- Współbieżny dostęp do zawartości pliku można synchronizować na poziomie całego pliku lub poszczególnych jego fragmentów (zajmowanie rekordów).
- Najczęściej dopuszcza się dwa rodzaje blokad
 - blokada współdzielona (shared lock, read lock)
 - blokada wyłączna (exclusive lock, write lock)



Synchronizacja dostępu do plików — zgodność blokad

	współdz.	wyłączna
współdz.	zgodne	wyklucz.
wyłączna	wyklucz.	wyklucz.