

wykład

Organizacja plików

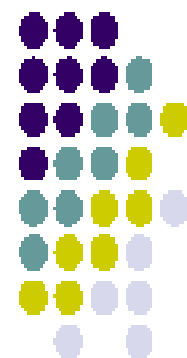




Plan wykładu

- Struktura przechowywania danych i organizacja rekordów w blokach
- Rodzaje organizacji plików
 - pliki nieuporządkowane
 - pliki uporządkowane
 - pliki haszowe

Wprowadzenie (1)

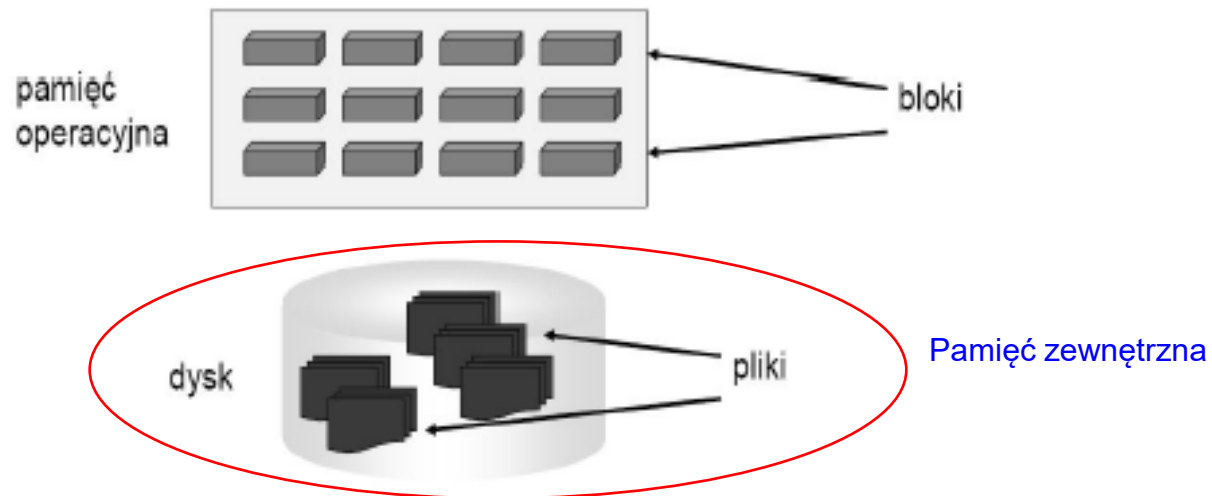


- Organizacja pliku
 - sposób uporządkowania rekordów w pliku przechowywanym na dysku
 - wspiera wykonywanie operacji na pliku
- Wybór odpowiedniej organizacji zależy od sposobu użytkowania danego pliku
 - wyszukiwanie rekordów opisujących zatrudnionych pracowników w porządku alfabetycznym \Rightarrow sortowanie pliku według nazwisk
 - wyszukiwanie rekordów opisujących zatrudnionych, których zarobki są w podanym zakresie \Rightarrow sortowanie nie jest właściwe
 - wybór odpowiedniej organizacji dla każdego pliku \Rightarrow administrator

Wprowadzenie (2)

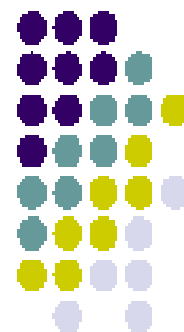


- Media fizyczne tworzą hierarchię pamięci składającą się z:
 - pamięci operacyjnej o organizacji blokowej
 - pamięci zewnętrznej o organizacji plikowej



Pamięć zewnętrzna → organizacja plikowa.
Pamięć operacyjna → organizacja blokowa.

Wprowadzenie (3)



- Trwale dane w BD są przechowywane w pamięci zewnętrznej:
 - ze względu na rozmiar danych
 - odporność pamięci zewnętrznej na awarie
 - niski koszt przechowywania
- Buforowanie bloków dyskowych
- Alokacja danych w plikach \Rightarrow rekordy
 - rekordy proste / złożone
 - rekordy o stałej / zmiennej długości
- Na poziomie dyskowym rekordy przechowywane w blokach dyskowych

Bufory bazy danych.

STRUKTURA PROSTA REKORDU : wartością każdego pola rekordu jest wartość elementarna (liczba, łańcuch znaków, data, ciąg bitów).

STRUKTURA ZŁOŻONA REKORDU: wartością pola rekordu jest inny rekord. Rekordy mogą mieć [stałą](#) długość lub [zmienną](#).

Stać długość rekordu długość oznacza, że rekord zawsze zajmuje tyle samo miejsca na dysku, niezależnie od rzeczywistych rozmiarów przechowywanych w nim danych.

Zmienna długość rekordu: rekordy o zmiennej długości przyjmują taki rozmiar jaki faktycznie przyjmują przechowywane w nich dane.

Współczynnik blokowania



- W bloku dyskowym lub bloku pamięci operacyjnej mieści się niecałkowita liczba rekordów
- Współczynnik blokowania \Rightarrow bfr (blocking factor)
 - rozmiar bloku B
 - rozmiar rekordu R

Współczynnik blokowania: $b_{fr} = \frac{B}{R}$ 1

Wolny obszar w bloku: $f_{space} = B - (b_{fr} \cdot R)$ 2

Współczynnik blokowania **bfr** jest określany jako największa liczba całkowita mniejsza od ilorazu rozmiaru bloku i rozmiaru rekordu.

Wolny obszar, jaki pozostaje w każdym z bloków, jest wyrażony wzorem 2:

$B - (bfr \cdot R)$.

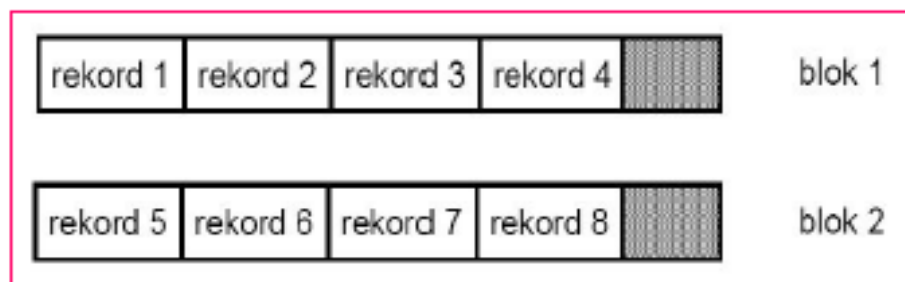
W przypadku rekordów o zmiennej długości jest to obszar minimalny.

Organizacje rekordów w blokach



organizacja dzielona
(spanned)

1



organizacja niepodzielna
(unspanned)

2

Rekordy w blokach: **dzielone** (ang. *spanned*) oraz **niepodzielne** (ang. *unspanned*).

Rekord dzielony: który cały nie mieści się w bloku jest dzielony, a jego części są przechowywane w kilku blokach.

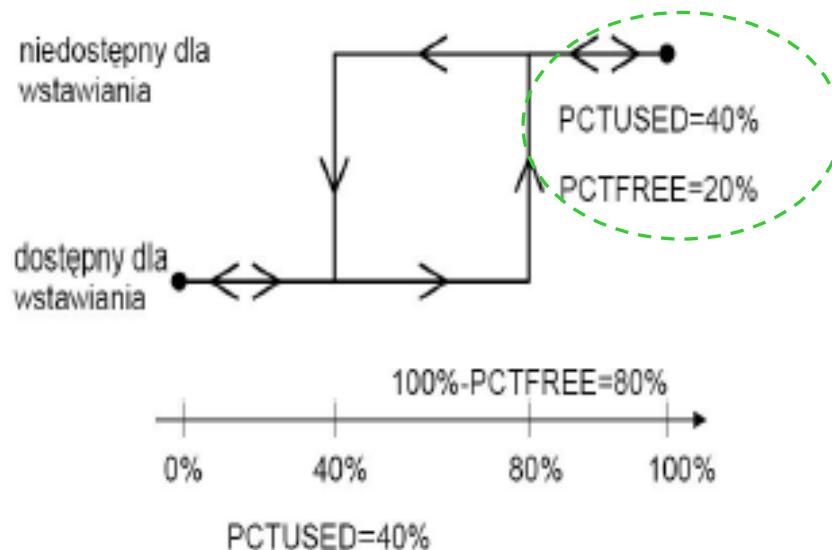
Rekord niepodzielny: rekord, który nie mieści się w całości w bloku jest przenoszony do takiego bloku, w którym zmieści się cały.

Organizacja niepodzielna powoduje, że w blokach pozostaje niewykorzystane miejsce.

zarządzanie rozmiarem bloku danych



Utrzymywanie wolnej pamięci w bloku dla potencjalnych modyfikacji



Definicja parametrów: **PCTFREE** oraz **PCTUSED**.

PCTFREE: określa ile procent rozmiaru bloku pozostanie wolne.

PCTUSED: określa, kiedy do bloku można wstawiać rekordy. Parametr ten jest wyrażony również procentowym rozmiarem bloku. Jeśli zajętość bloku przewyższa wartość PCTUSED, wówczas blok nie jest wykorzystywany do wstawiania nowych rekordów. Jeśli natomiast zajętość bloku jest niższa niż PCTUSED, wówczas do bloku można wstawiać rekordy.

operacje na plikach



- Operacje dostępu do pojedynczego rekordu (record-at-a-time)
 - wyszukiwanie: Find, FindFirst, FindNext
 - usuwanie: Delete
 - aktualizacja: Update
 - wstawianie: Insert
- Operacje dostępu do zbioru rekordów (set-at-a-time)
 - wyszukiwanie wszystkich rekordów spełniających kryterium: FindAll
 - wyszukiwanie z sortowaniem: FindOrdered
 - reorganizacja pliku: Reorganize

Ponieważ rekordy są fizycznie przechowywane w plikach, więc dostęp do rekordów musi być realizowany za pomocą dedykowanych i zoptymalizowanych operacji.

Wyróżnia się operacje na **pojedynczym rekordzie** (ang. record-at-a-time) i operacje na **zbiorze rekordów** (ang. set-at-a-time).

model kosztów



Model kosztów

- Notacja i założenia:
 - $N \Rightarrow$ liczba bloków
 - każdy blok zawiera R rekordów
 - średni czas odczytu/zapisu bloku dyskowego wynosi D
 - średni czas przetwarzania rekordu wynosi C
 - dla plików haszowych czas obliczenia wartości funkcji haszowej wynosi H
- Typowe wartości wynoszą:
 - $D = 15$ ms
 - C i H od 1 do 10 μ s
 - dominuje koszt I/O

Operacja na pliku posiada swój tzw. **koszt**, który jest zależny od organizacji wewnętrznej pliku. Koszt jest konkretną wartością, której miarą może być np. [czas wykonania, liczba dostępów do dysku](#).

Koszt jest wartością wynikającą z tzw. **modelu kosztów** (ang. *cost model*).

rodzaje organizacji plików



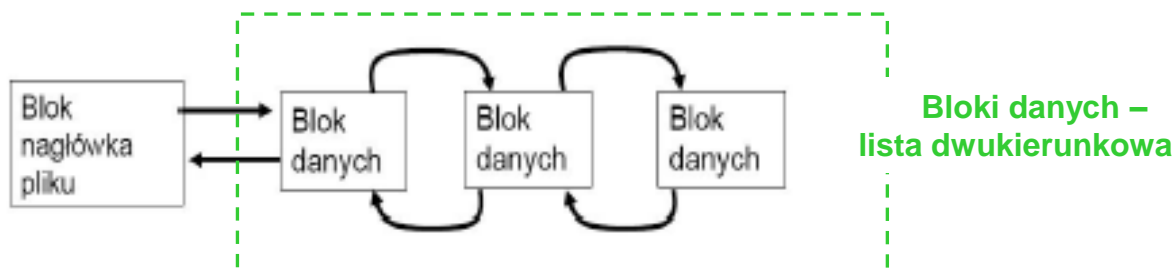
- 1 • Pliki nieuporządkowane (unordered files, heap files)
- 2 • Pliki uporządkowane (ordered files)
- 3 • Pliki haszowe (hash files)

pliki nieuporządkowane

1



- Nagłówek pliku zawierający wskaźnik do bloku danych
- Blok danych zawiera wskaźnik do bloku następnego i poprzedniego
- Rekordy wstawiane na koniec pliku



Charakterystyka

- Efektywne wstawianie pojedynczych rekordów i dużych zbiorów rekordów
- Efektywne pozostałe operacji w przypadku plików o rozmiarze kilku bloków
- Struktura właściwa dla odczytu wszystkich rekordów
- Struktura stosowana z innymi strukturami dostępu do danych (np. indeksy)

Plik nieuporządkowany: rekordy są przechowywane w kolejności ich wstawiania. Nagłówek pliku zawiera: wskaźnik do pierwszego bloku danych.

Bloki danych tworzą listę dwukierunkową.

pliki nieuporządkowane – operacje (1)



1

- Podstawową organizacją pliku nieuporządkowanego jest stóg (ang. heap)
- Operacje:
 - wstawianie rekordu
 - rekord jest wstawiany do ostatniego bloku pliku; blok ten jest zapisywany na dysk
 - koszt = $2D+C$
 - wyszukiwanie rekordu
 - konieczność liniowego przeszukiwania wszystkich bloków
 - średni koszt = $0.5N(D+RC)$ 1
 - maksymalny koszt = $N(D+RC)$ (przejrzenie całego pliku) 2

pliki nieuporządkowane – operacje (2)



1

– przeglądanie pliku

- koszt = $N(D + RC)$ 1
- odczyt N stron z kosztem D
- dla każdej strony przetworzyć R rekordów z kosztem C

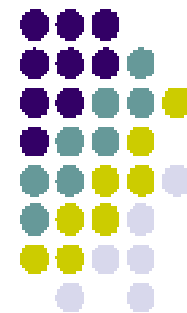
– wyszukiwanie z przedziałem wartości

- odczyt wszystkich bloków
- koszt = $N(D + RC)$ 2

– usuwanie rekordu

- liniowe przeszukiwanie i zapis bloku na dysk
- koszt wyszukania + koszt $(C + D)$
 - średnio: $0.5N(D+RC)+(C+D)$ 3
 - maksymalnie: $N(D+RC)+(C+D)$ 4
- pozostaje zwolniona pamięć \Rightarrow konieczność okresowej reorganizacji pliku

pliki nieuporządkowane – operacje (3)



1

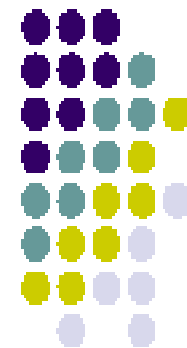
– sortowanie

- trudne
- stosowane tzw. sortowanie zewnętrzne
 - plik sortowany fragmentami mieszczącymi się w pamięci operacyjnej
 - fragmenty są łączone w większe w kolejnych przebiegach algorytmu sortowania

Sortowanie → wykonywane w pamięci operacyjnej. Techniką sortowania stosową najczęściej **Sortowanie zewnętrzne** (ang. *external sorting*) → sortowanie pliku fragmentami, które mieszczą się w pamięci operacyjnej. Każdy posortowany fragment jest w drugiej fazie sortowania łączony z innymi fragmentami. Łączenie to wymaga zwykle wielu przebiegów.

pliki nieuporządkowane – charakterystyka

1



- Efektywne wstawianie pojedynczych rekordów i dużych zbiorów rekordów
- Efektywne pozostałe operacji w przypadku plików o rozmiarze kilku bloków
- Struktura właściwa dla odczytu wszystkich rekordów
- Struktura stosowana z innymi strukturami dostępu do danych (np. indeksy)

pliki uprządkowane

Struktura organizacji plików uporządkowanych

2



pole porządkujące

Aaron Brian			
Abbott Jim	blok 1		
Acosta Phil			

Adams Jim			
Adams Robin	blok 2		
Adler John			

Winslet Kathy			
Zobel Mick	blok N		
Zezula Pavol			

W pliku uporządkowanym rekordy składowane są uporządkowane wg wartości tzw. pola porządkującego (ang. *ordering field*).

W przykładowym pliku ze slajdu, rekordy zostały uporządkowane wg wartości **nazwiska i imienia**.

pliki uporządkowane – operacje (1)

2



Operacje:

– przeglądanie pliku

- koszt = $N(D+RC)$

1

- ponieważ wszystkie strony muszą być odczytane

– wyszukiwanie rekordu

- wyszukiwanie binarne

- $koszt = D \cdot \log_2 B + C \cdot \log_2 R$

2

– wyszukiwanie z przedziałem wartości

- koszt wyszukania pierwszego rekordu + koszt selekcji zbioru rekordów

$$koszt = D \cdot \log_2 B + C \cdot \log_2 R + ND$$

3

pliki uporządkowane – operacje (2)

2



Operacje:

– wstawianie rekordu

- koszt wyszukania miejsca wstawienia rekordu
- koszt przepisania średnio połowy rekordów
- koszt całkowity = $2 \cdot (0.5N(D+RC))$ ①

– usuwanie rekordu

- koszt znalezienia usuwanego rekordu
- koszt przepisania średnio połowy rekordów
- identyczny jak koszt wstawiania
- koszt całkowity = $2 \cdot (0.5N(D+RC))$ ②

plik uporządkowany

- rozwiązanie problemu wstawiania rekordów

2



- Przesuwanie średnio połowy rekordów \Rightarrow nieefektywne
- Pozostawienie wolnej pamięci w każdym bloku na wstawiane rekordy
 - wstawienie rekordu wymaga przesunięcia rekordów tylko w ramach bieżącego bloku
- Tworzenie nieuporządkowanego pliku rekordów, a następnie łączenie go z plikiem głównym
 - efektywne wstawianie
 - nieefektywne wyszukiwanie

Pierwsze rozwiązanie: *przesuwanie średnio połowy rekordów w pliku, w celu zagwarantowania właściwego porządku rekordów w pliku.*

Drugie rozwiązanie: pozostawienie w każdym bloku dyskowym pustego miejsca na wstawiane rekordy. W tym przypadku przesunięcia rekordów należy dokonać tylko w tym bloku, do którego jest wstawiany rekord.

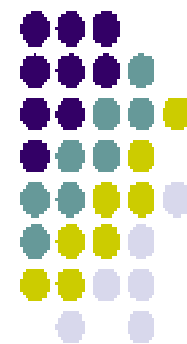
Trzecie rozwiązanie: wykorzystanie nieuporządkowanego tymczasowego pliku, tzw.: nadmiarowego (ang. *overflow file*) lub transakcyjnego (ang. *transaction file*).

plik uporządkowany

- modyfikowanie rekordu

PROCEDURA REALIZACJI

2



- Znalezienie i odczyt modyfikowanego rekordu do bufora
 - połowienie binarne dla warunku na polu porządkującym
 - przeszukanie całego pliku dla innego warunku
- Modyfikowanie wartości atrybutu nieporządkującego
 - zmodyfikowanie w buforze i zapis na dysk w to samo miejsce
- Modyfikowanie wartości atrybut porządkującego
 - zmiana pozycji rekordu w pliku
 - usunięcie rekordu + wstawienie nowego

Modyfikowanie rekordu wymaga jego znalezienia w pliku i odczytania do bufora.

plik uporządkowany – zalety i wady

2



ZALETY:

- Efektywny odczyt rekordów w kolejności pola porządkującego \Rightarrow posortowane
- Znalezienie następnego rekordu, według określonego porządku, jest bardzo proste
- Metoda połowienia binarnego do wyszukiwania rekordu z warunkiem opartym o pole porządkujące

WADY:

- Uporządkowanie pliku jest nieprzydatne, gdy wyszukiwanie jest realizowane według wartości pola nie porządkującego pliku danych,
- Kosztowne wstawianie i usuwanie rekordów ze względu na konieczność zachowania porządku w pliku

Zalety: Operacja odczytu i sortowania \rightarrow efektywna, znalezienie następnego rekordu, według określonego porządku \rightarrow bardzo proste.

Wady: Uporządkowanie pliku jest nieprzydatne, gdy wyszukiwanie jest realizowane według wartości pola nie porządkującego pliku danych.

Pliki nieuporządkowane



❑ Wstawianie rekordu

➤ $K = 2D + C = 2 \cdot 15 \cdot 10^{-3} + 5 \cdot 10^{-6} = \text{???? } s$

❑ Wyszukiwanie rekordu

➤ $K_{\text{śr}} = 0,5 \cdot N(D + RC) = 0,5 \cdot 3000(15 \cdot 10^{-3} + 30000 \cdot 5 \cdot 10^{-6}) = \text{??? } s$

➤ $K_{\text{max}} = N(D + RC) = 3000(15 \cdot 10^{-3} + 30000 \cdot 5 \cdot 10^{-6}) = \text{??? } s$

❑ Przeglądanie pliku

➤ $K = N(D + RC) = 3000(15 \cdot 10^{-3} + 30000 \cdot 5 \cdot 10^{-6}) = \text{??? } s$

❑ Wyszukiwanie z przedziałem wartości

➤ $K = N(D + RC) = 3000(15 \cdot 10^{-3} + 30000 \cdot 5 \cdot 10^{-6}) = \text{??? } s$

❑ Usuwanie rekordu

➤ $K_{\text{śr}} = 0,5 \cdot N(D + RC) + (C + D) = 0,5 \cdot 3000(15 \cdot 10^{-3} + 30000 \cdot 5 \cdot 10^{-6}) + (15 \cdot 10^{-3} + 5 \cdot 10^{-6}) = 0,5 \cdot 495 \approx \text{??? } s$

➤ $K_{\text{max}} = N(D + RC) + (C + D) = 3000(15 \cdot 10^{-3} + 30000 \cdot 5 \cdot 10^{-6}) + (15 \cdot 10^{-3} + 5 \cdot 10^{-6}) \approx \text{??? } s$

Pliki uporządkowane



❑ Wstawianie rekordu

➤ $K = 2 \cdot (0,5 \cdot N(D + RC)) = 2 \cdot (0,5 \cdot 3000(15 \cdot 10^{-3} + 30000 \cdot 5 \cdot 10^{-6})) = 2 \cdot 0,5 \cdot 495 = ??? \text{ s}$

❑ Wyszukiwanie rekordu

➤ $K = D \cdot \log_2 B + C \cdot \log_2 R = 15 \cdot 10^{-3} \cdot \log_2 1024 + 5 \cdot 10^{-6} \cdot \log_2 30000 \approx ??? \text{ s}$

❑ Przeglądanie pliku

➤ $K = N(D + RC) = 3000(15 \cdot 10^{-3} + 30000 \cdot 5 \cdot 10^{-6}) = ??? \text{ s}$

❑ Wyszukiwanie rekordu z przedziałem wartości

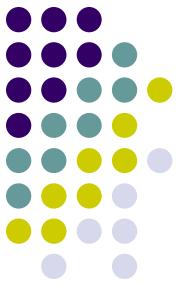
➤ $K = D \cdot \log_2 B + C \cdot \log_2 R + ND = 15 \cdot 10^{-3} \cdot \log_2 1024 + 5 \cdot 10^{-6} \cdot \log_2 30000 + 3000 \cdot 15 \cdot 10^{-3} \approx ??? \text{ s}$

❑ Usuwanie rekordu

➤ $K = 2 \cdot (0,5 \cdot N(D + RC)) = 2 \cdot (0,5 \cdot 3000(15 \cdot 10^{-3} + 30000 \cdot 5 \cdot 10^{-6})) = 2 \cdot 0,5 \cdot 495 = ??? \text{ s}$

Pliki uporządkowane

Pliki nieuporządkowane



❑ Wstawianie rekordu

- $K = 2 \cdot 0,5 \cdot N(D + RC) = 2 \cdot 0,5 \cdot 3000(15 \cdot 10^{-3} + 30000 \cdot 5 \cdot 10^{-6}) = 2 \cdot 0,5 \cdot 495 = ??? \text{ s}$

❑ Wyszukiwanie rekordu

- $K = D \cdot \log_2 B + C \cdot \log_2 R = 15 \cdot 10^{-3} \cdot \log_2 1024 + 5 \cdot 10^{-6} \cdot \log_2 30000 \approx ??? \text{ s}$

❑ Przeglądanie pliku

- $K = N(D + RC) = 3000(15 \cdot 10^{-3} + 30000 \cdot 5 \cdot 10^{-6}) = ??? \text{ s}$

❑ Wyszukiwanie rekordu z przedziałem wartości

- $K = D \cdot \log_2 B + C \cdot \log_2 R + ND = 15 \cdot 10^{-3} \cdot \log_2 1024 + 5 \cdot 10^{-6} \cdot \log_2 30000 + 3000 \cdot 15 \cdot 10^{-3} \approx ??? \text{ s}$

❑ Usuwanie rekordu

- $K = 2 \cdot 0,5 \cdot N(D + RC) = 2 \cdot 0,5 \cdot 3000(15 \cdot 10^{-3} + 30000 \cdot 5 \cdot 10^{-6}) = 2 \cdot 0,5 \cdot 495 = ??? \text{ s}$

❑ Wstawianie rekordu

- $K = 2D + C = 2 \cdot 15 \cdot 10^{-3} + 5 \cdot 10^{-6} = ??? \text{ s}$

❑ Wyszukiwanie rekordu

- $K_{sr} = 0,5 \cdot N(D + RC) = 0,5 \cdot 3000(15 \cdot 10^{-3} + 30000 \cdot 5 \cdot 10^{-6}) = ??? \text{ s}$
- $K_{max} = N(D + RC) = 3000(15 \cdot 10^{-3} + 30000 \cdot 5 \cdot 10^{-6}) = ??? \text{ s}$

❑ Przeglądanie pliku

- $K = N(D + RC) = 3000(15 \cdot 10^{-3} + 30000 \cdot 5 \cdot 10^{-6}) = ??? \text{ s}$

❑ Wyszukiwanie z przedziałem wartości

- $K = N(D + RC) = 3000(15 \cdot 10^{-3} + 30000 \cdot 5 \cdot 10^{-6}) = ??? \text{ s}$

❑ Usuwanie rekordu

- $K_{sr} = 0,5 \cdot N(D + RC) + (C + D) = 0,5 \cdot 3000(15 \cdot 10^{-3} + 30000 \cdot 5 \cdot 10^{-6}) + (15 \cdot 10^{-3} + 5 \cdot 10^{-6}) \approx ??? \text{ s}$
- $K_{max} = N(D + RC) + (C + D) = 3000(15 \cdot 10^{-3} + 30000 \cdot 5 \cdot 10^{-6}) + (15 \cdot 10^{-3} + 5 \cdot 10^{-6}) \approx ??? \text{ s}$

plik haszowy

3



Plik o strukturze wykorzystującej technikę haszowania nazywa się plikiem haszowym (bezpośrednim)

- Porządek rekordów w pliku określony na podstawie tzw. pola haszowego
- Koncepcja
 - zdefiniowanie funkcji haszowej (ang. hash function) z argumentem, którym jest wartość pola haszowego
 - wynikiem funkcji haszowej jest adres bloku dyskowego, w którym powinien znaleźć się dany rekord
- Haszowanie
 - wewnętrzne
 - zewnętrzne

Koncepcja organizacji tego pliku polega na zdefiniowaniu **funkcji haszowej** (ang. *hash function*). W praktyce stosuje się dwie techniki haszowania, tj. **haszowanie wewnętrzne i haszowanie zewnętrzne**.

haszowanie wewnętrzne

3



Dana jest tablica rekordów o M szczelinach, których adresy odpowiadają indeksom tablicy haszowej.

indeksy tablicy
haszowej

Indeks tablicy: $0 \Rightarrow M-1$

Funkcja haszowa

$H(K=\text{wartość pola haszowego}) \rightarrow \{0, \dots, M-1\}$

Najczęściej spotykana funkcja haszowa

$H(K)=K \text{ MOD } M$

	Id_prac	Nazwisko	Etat	Płaca
0	450	Nowak	kierownik	2000
1	120	Kuzio	portier	1000
2	220	Misiek	z-ca kier.	1800
M-2	100	Dziubek	dyrektor	5000
M-1	110	Gulczas	portier	800

Koncepcja haszowania wewnętrznego

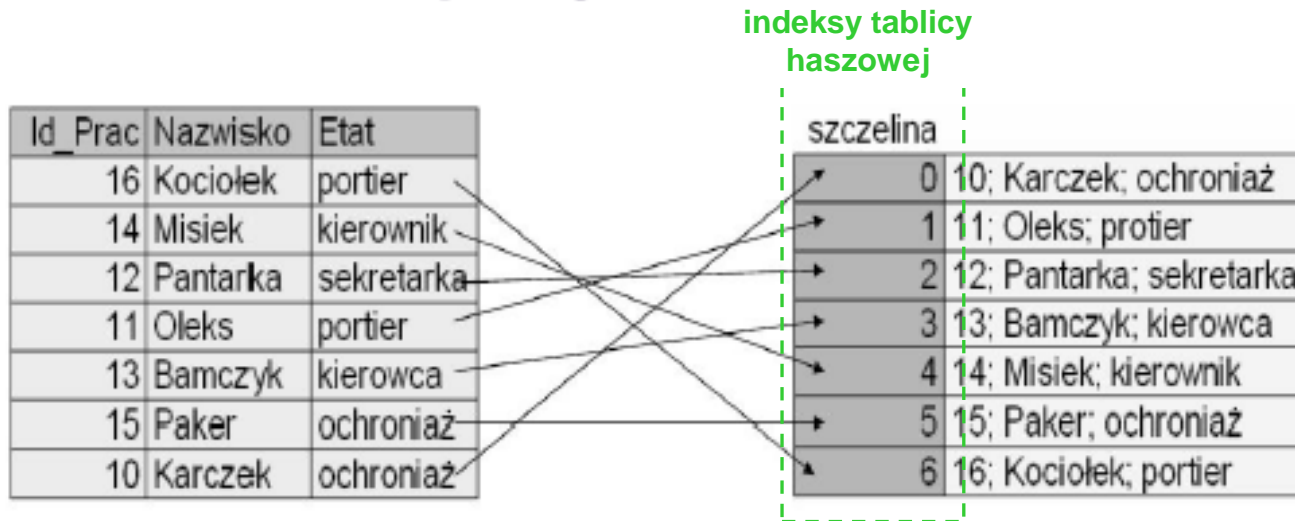
Dana jest tablica rekordów o M szczelinach. Adresy szczelin odpowiadają indeksom tablicy haszowej. Indeksy te przyjmują wartości od 0 do M-1.

Funkcja haszowa ma postać: $H(K=\text{wartość pola haszowego}) \rightarrow \{0, \dots, M-1\}$.

Funkcja ta transformuje wartość pola haszowego do liczby całkowitej z zakresu od 0 do M-1.

haszowanie przykład

3



$$H(\text{Id_Prac}) = \text{Id_Prac} \bmod 10$$

Przykład haszowania wewnętrznego: rekordy pracowników z polami:

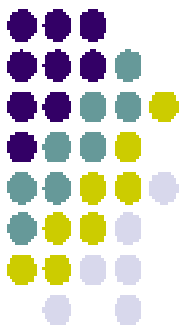
Id_Prac, Nazwisko, Etat.

Przyjmijmy tablicę haszową ze szczelinami o numerach od 0 do 6.

Definicja funkcji haszowej: **$H(\text{Id_Prac}) = \text{Id_Prac} \bmod 10$**

proces haszowania

3



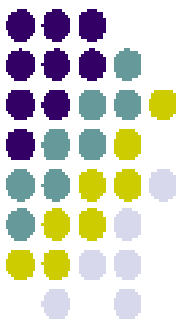
Proces haszowania składa się z dwóch kroków:

1. transformacji (pola nienumeryczne są transformowane do liczb całkowitych)
2. haszowania (przykładowy algorytm transformacji)

```
temp := 1;  
for i = 1 to 20 do  
temp := temp * code(K[i])  
hash_address := temp MOD M;
```

kolizja - alternatywa

3



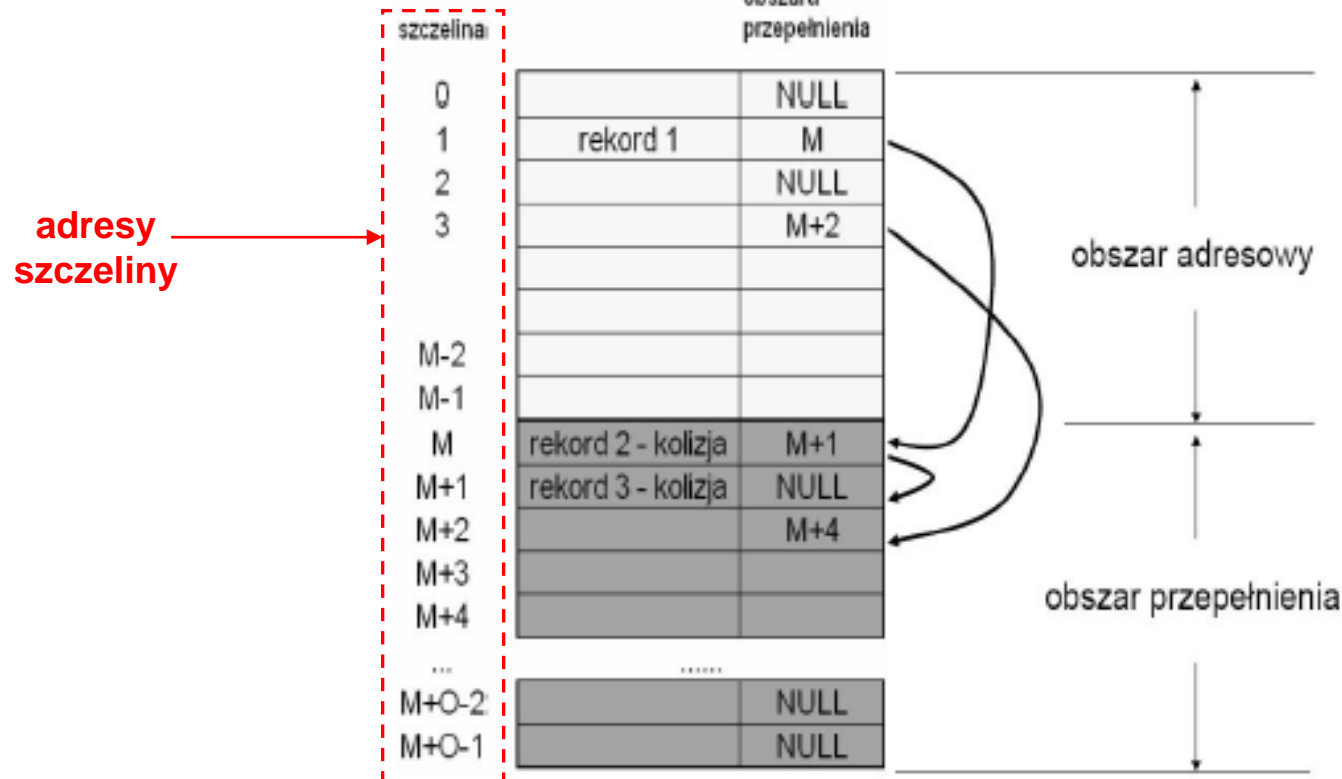
- Kolizja \Rightarrow wartość funkcji haszowej dla danej wartości pola haszowego nowego rekordu odpowiada zajętemu już adresowi szczeliny
- Źródło kolizji \Rightarrow funkcja haszowa generująca te same adresy szczeliny dla różnych wartości atrybutu haszowego
- Rozwiązanie kolizji \Rightarrow procedura znajdowania innej lokalizacji dla wprowadzanego rekordu

METODY ROZWIĄZYWANIA KOLIZJI

- Adresowanie otwarte (open addressing) – następna wolna lokalizacja
- Łańcuchowanie (chaining) – wskaźniki do nowych lokalizacji
- Haszowanie wielokrotne (multiple hashing) – nowa funkcja haszowa

łańcuchowanie

3



Technika łańcuchowania polega na przechowywaniu w szczelinie dodatkowo wskaźnika do tzw. obszaru przepełnienia (ang. *overflow space*). Służy on do przechowywania wszystkich rekordów ulegających kolizji w danej szczelinie. Rekordy w obszarze przepełnienia tworzą listę.

haszowanie zewnętrzne (1)

3



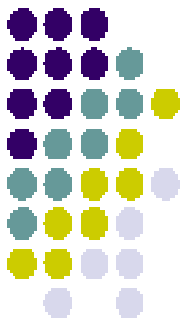
- Wartościami tablicy haszowej są adresy logicznych obszarów dyskowych (LOD) (block buckets)
- Liczba LOD jest stała i równa liczbie szczelin w tablicy haszowej \Rightarrow jest to tzw. haszowanie statyczne (static hashing)
- LOD jest albo pojedynczym blokiem dyskowym albo zbiorem kolejnych (leżących obok siebie) bloków dyskowych
- Funkcja haszowa odwzorowuje wartość atrybutu haszowego w numer LOD
- Plik dyskowy zawiera tablicę konwersji numerów LOD w fizyczne adresy bloków dyskowych

W haszowaniu zewnętrznym wartościami tablicy haszowej są adresy logicznych obszarów dyskowych (LOD) (ang. *block buckets*).

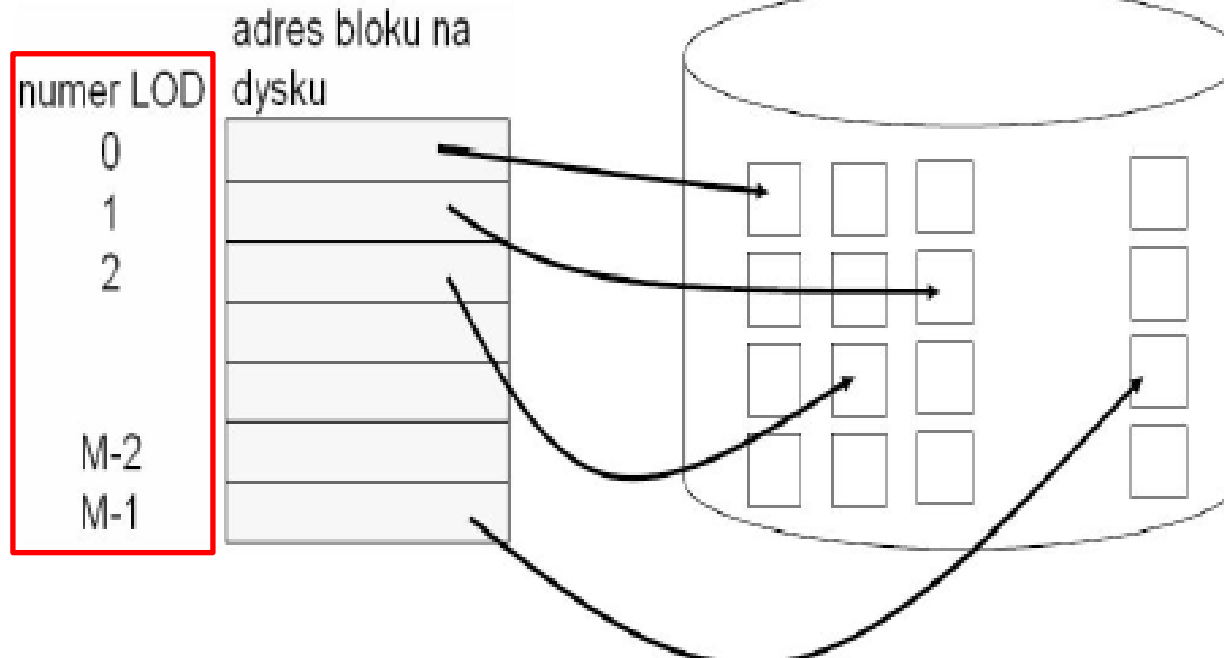
Funkcja haszowa odwzorowuje wartość atrybutu haszowego w numer LOD. Plik dyskowy zawiera tablicę konwersji numerów LOD w fizyczne adresy bloków dyskowych.

haszowanie zewnętrzne (2)

3

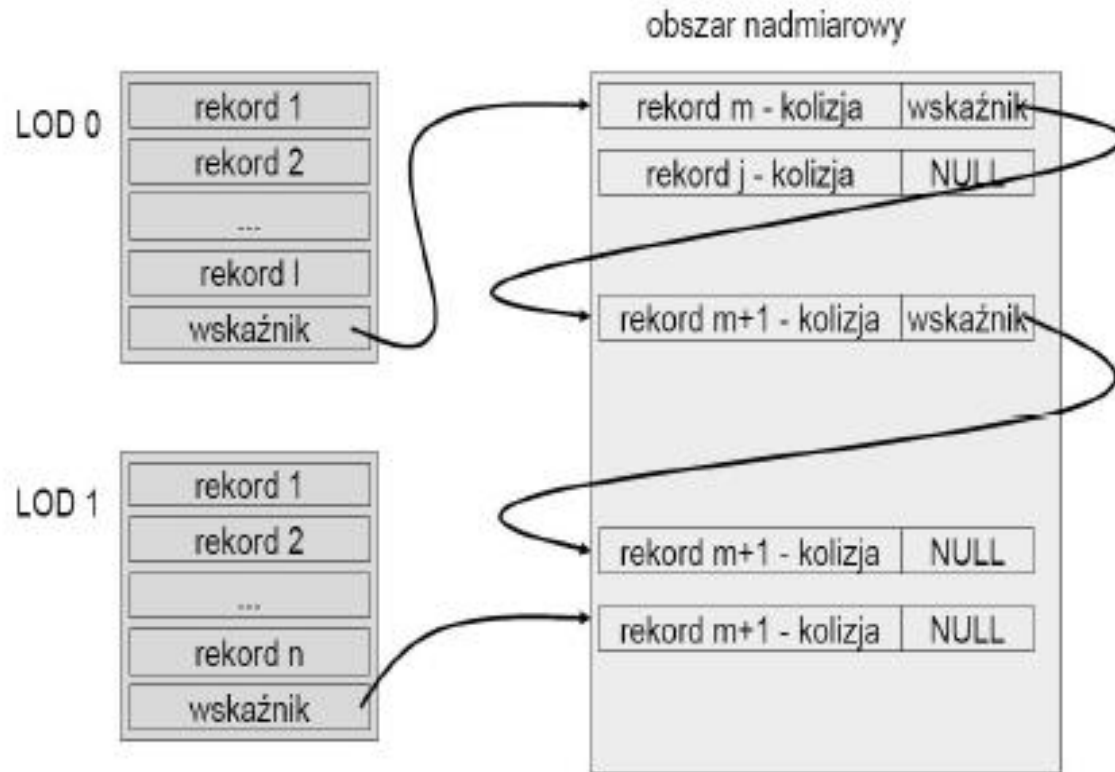


tablica konwersji



haszowanie zewnętrzne - kolizje

3



W haszowaniu zewnętrznym → kolizje rzadsze, (ten sam LOD, którego numer jest wynikiem działania funkcji haszowej, może pomieścić wiele rekordów).

Kolizja może jednak wystąpić po zapełnieniu się wszystkich bloków dyskowych wchodzących w skład danego LOD.

haszowanie zewnętrzne - operacje (1)

3



- Poszukiwanie rekordu z warunkiem nałożonym na pole inne niż haszowe
 - przeszukanie całego pliku i obszaru nadmiarowego
- Poszukiwanie rekordu z warunkiem nałożonym na pole haszowe
 - funkcja haszowa
 - w przypadku kolizji przeszukanie obszaru nadmiarowego
- Usunięcie rekordu z LOD
 - wyszukanie rekordu (funkcja haszowa + tablica konwersji) i zwolnienie szczeliny
 - przesunięcie pierwszego rekordu z obszaru nadmiarowego do LOD
- Usunięcie rekordu z obszaru nadmiarowego
 - wyszukanie rekordu (funkcja haszowa) + przeszukanie listy rekordów w obszarze nadmiarowym
 - zwolnienie szczeliny
 - utrzymywanie listy wolnych szczelin

Poszukiwanie w pliku haszowym rekordu z warunkiem nałożonym na pole inne niż haszowe wymaga przeszukania całego pliku i obszaru nadmiarowego.

Usunięcie rekordu z pliku haszowego przebiega w dwóch wariantach.

haszowanie zewnętrzne - operacje (1)

3



- Poszukiwanie rekordu z warunkiem nałożonym na pole inne niż haszowe
 - przeszukanie całego pliku i obszaru nadmiarowego
- Poszukiwanie rekordu z warunkiem nałożonym na pole haszowe
 - funkcja haszowa
 - w przypadku kolizji przeszukanie obszaru nadmiarowego
- Usunięcie rekordu z LOD
 - wyszukanie rekordu (funkcja haszowa + tablica konwersji) i zwolnienie szczeliny
 - przesunięcie pierwszego rekordu z obszaru nadmiarowego do LOD
- Usunięcie rekordu z obszaru nadmiarowego
 - wyszukanie rekordu (funkcja haszowa) + przeszukanie listy rekordów w obszarze nadmiarowym
 - zwolnienie szczeliny
 - utrzymywanie listy wolnych szczelin

W wariancie drugim, usuwany rekord znajduje się w obszarze nadmiarowym. Jego usunięcie polega na znalezieniu szczeliny (z wykorzystaniem funkcji haszowej i tablicy konwersji) w LOD.

haszowanie zewnętrzne - operacje (3)

3



- Wstawienie rekordu
 - odczyt adresu szczeliny (funkcja haszowa)
 - w przypadku kolizji zaalokowanie szczeliny w obszarze nadmiarowym
- Zmodyfikowanie wartości pola haszowego
 - odczytanie rekordu (funkcja haszowa)
 - rekord zmienia szczelinę
- Usunięcie rekordu + wstawienie rekordu z nową wartością
- Zmodyfikowanie wartości pola nie-haszowego
 - odczytanie rekordu (funkcja haszowa)
 - zmodyfikowanie wartości
 - rekord nie zmienia szczeliny

Wstawienie rekordu do pliku haszowego polega na odczytaniu adresu szczeliny z wykorzystaniem funkcji haszowej i zapisaniu rekordu do tej szczeliny.

Zmodyfikowanie wartości pola haszowego polega na odczytaniu rekordu z wykorzystaniem funkcji haszowej.

funkcja haszowa

3



- Cechą dobrej funkcji haszowej jest zapewne równomiernego rozkładu rekordów w obrębie przestrzeni adresowej tablicy haszowej
- Zalecany rozmiar tablicy haszowej
 $r/M \in (0,7 \div 0,9)$, $r \Rightarrow$ liczba rekordów, $M \Rightarrow$ liczba bloków

Charakterystyka

- Problem porządkowania pliku oraz wyszukiwania rekordów w porządku wartości pola haszowego
- Problem stałego rozmiaru przestrzeni adresowej przydzielonej plikowi – częste kolizje

Praktyka pokazuje, że tablica haszowa powinna być zajęta w 70% - 90%
Pliki haszowe są nieefektywne dla operacji odczytu danych z ich porządkowaniem zgodnie z wartością pola haszowego.



KONIEC WYKŁADU