

ORGANIZACJA LOGICZNA PLIKÓW

Wprowadzenie (1)

- Organizacja pliku określa sposób uporządkowania rekordów w pliku przechowywanym na dysku
- Różne sposoby organizacji rekordów wspomagają wykonywanie różnych operacji na pliku
 - Przykładowo, jeśli chcemy wyszukiwać rekordy opisujące zatrudnionych pracowników w porządku alfabetycznym, sortowanie pliku według nazwisk jest dobrą organizacją pliku. Z drugiej strony, jeśli chcemy wyszukiwać rekordy opisujące zatrudnionych, których zarobki są w podanym zakresie, sortowanie rekordów pracowników według nazwisk nie jest właściwą organizacją pliku

Wprowadzenie (2)

- SZBD wspiera kilka organizacji plików, a zadaniem DBA jest wybranie odpowiedniej organizacji dla każdego pliku.
- Wybór odpowiedniej organizacji zależy od sposobu użytkowania danego pliku
- Media fizyczne tworzą hierarchię pamięci składającą się z dwóch kategorii:
 - pamięć operacyjna
 - pamięć zewnętrzną

Wprowadzenie (3)

- Dane w bazie danych są pamiętane w długim odcinku czasu w pamięci zewnętrznej z trzech powodów:
 - ze względu na rozmiar bazy danych
 - odporność pamięci zewnętrznej na awarie
 - koszt jednostkowy
- Buforowanie bloków dyskowych.
- Alokacja plików na dysku.
 - rekordy - typy rekordów i typy danych
 - rekordy o stałej i zmiennej długości

Współczynnik blokowania

- Współczynnik blokowania - bfr
 - rozmiar bloku B; rozmiar rekordu R
 - współczynnik blokowania $bfr = \lfloor (B/R) \rfloor$
 - wolny obszar B - (bfr*R) rekordów

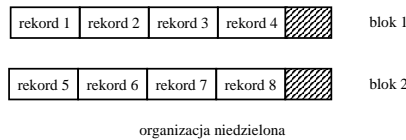
Typy organizacji (1)

- Organizacja pliku:
 - organizacja dzielona (ang. spanned)



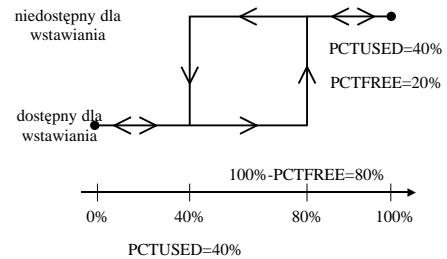
Typy organizacji (2)

- Organizacja pliku:
 - organizacja niedzielona (ang. spanned)



Zarządzanie danymi w ramach bloku danych

Utrzymywanie wolnej pamięci w bloku dla potencjalnych modyfikacji



Operacje na plikach

- Operacje typu *record-at-a-time*:
 - wyszukiwania: *FindFirst*, *Read*, *FindNext*
 - usuwania: *Delete*
 - aktualizacji: *Update*
 - wstawiania: *Insert*
- Operacje typu *set-at-a-time*:
 - FindAll*
 - FindOrdered*
 - Reorganize*

Model kosztów

- Notacja i założenia:
 - N oznacza ilość stron, każda strona zawiera R rekordów,
 - średni czas odczytu/zapisu bloku dyskowego wynosi D ,
 - średni czas przetwarzania rekordu (np., porównanie wartości atrybutu ze stałą) wynosi C ,
 - w przypadku plików haszowych stosujemy funkcję haszową odwzorowującą wartości rekordów na liczby naturalne; 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

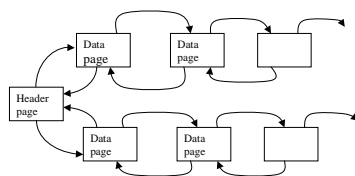
Plik nieuporządkowany

- Podstawową organizacją pliku nieuporządkowanego jest **stóg** (ang. heap)
- Operacje:
 - wstawianie rekordów**: rekord jest wstawiany do ostatniego bloku pliku; blok ten jest zapisywany na dysk - koszt wynosi $2D+C$
 - wyszukiwanie rekordu**: konieczność **liniowego przeszukiwania** wszystkich bloków - koszt wynosi $0.5N(D+RC)$. Jeżeli nie ma rekordów spełniających warunek selekcji system musi przejrzeć cały plik

Plik nieuporządkowany

- przeglądanie pliku**: - koszt wynosi $N(D+RC)$ ponieważ musimy odczytać N stron (D koszt odczytu strony), dla każdej strony, przetworzyć R rekordów (C per rekord)
- wyszukiwanie z przedziałem wartości**: konieczność odczytu wszystkich bloków - koszt wynosi $N(D+RC)$.
- usuwanie rekordu**: liniowe przeszukiwanie i zapis bloku na dysk; tj. koszt wyszukania + koszt ($C+D$); pozostaje zwolniona pamięć - konieczność okresowej reorganizacji pliku w celu odzyskania pamięci.
- problem sortowania pliku** - tzw. **sortowanie zewnętrzne**

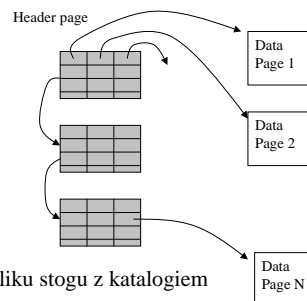
Plik nieuporządkowany



Stóg – podwójna lista stron:

- lista stron z wolną przestrzenią
- lista stron pełnych

Plik nieuporządkowany



Struktura pliku stogu z katalogiem

Wolna przestrzeń dyskowa jest wskazywana przez pojedynczy bit (na stronę)

Plik uporządkowany

1	Aaron Brian			
	Abbott Jim			
	Acosta Phil			
2	Adams Jim			
	Adams Robin			
	Adler John			
N	Winslet Kathy			
	Zobel Mick			
	Zze zula Paval			

Plik uporządkowany

- Rekordy pliku są porządkowane według **poła porządkującego** (ang. ordering field). Prowadzi to do organizacji typu **plików uporządkowanych** lub **sekwencyjnych** (ang. ordered) lub (ang. sequential)
- Operacje:
 - **przeglądanie pliku**: koszt wynosi $N(D+RC)$, ponieważ wszystkie strony muszą być odczytane
 - **wyszukiwanie rekordu**: wyszukiwanie binarne - koszt wynosi $D \log_2 B + C \log_2 R$

Plik uporządkowany

- **wyszukiwanie z przedziałem wartości**: koszt jest równy kosztowi wyszukiwania rekordu i selekcji zbioru rekordów.
- **wstawianie rekordu**: koszt ten odpowiada kosztowi wyszukiwania rekordu (określenie pozycji rekordu) oraz przepisania połowy rekordów - $2 * (0.5N(D+RC))$;
- **usuwanie rekordu**: koszt usuwania odpowiada kosztowi wstawiania tj. $2 * (0.5N(D+RC))$;

Plik uporządkowany

Zalety:

- dla operacji porządkowania wyniku zapytania według pola porządkującego plik jest już uporządkowany
- znalezienie następnego rekordu, według określonego porządku, jest bardzo proste
- jeżeli warunek wyszukiwania rekordu oparty na polu porządkującym w celu wyszukiwania danych można zastosować szybką metodę połowienia binarnego pliku

Plik uporządkowany

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
 - podejście z przesuwaniem pliku (średnio 1/2 rekordów do przesunięcia)
 - pozostawienie wolnej pamięci w bloku
 - tworzenie nieuporządkowanego pliku rekordów, tzw. **overflow** lub **transaction file**, a następnie łączenie go z plikiem głównym
- modyfikowanie rekordów

Pliki haszowe

- Plik o strukturze wykorzystującej technikę haszowania nosi nazwę pliku **haszowego** lub **bezpośredniego**.
- Podstawą określającą porządek rekordów w pliku jest jedno z pól rekordów zwane **polem haszowym**.
- Idea polega na zdefiniowaniu **funkcji haszowej** (ang. hash function), która dla argumentu równego wartości pola haszowego wyznacza wynik, który jest adresem bloku dyskowego, w którym powinien znaleźć się dany rekord

Haszowanie wewnętrzne

	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	Gukzas	z-ca portiera	800

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

Indeks tablicy: 0 – M-1

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

Najczęściej spotykaną funkcją haszową jest funkcja

$$h(K)=K \bmod M$$

Haszowanie wewnętrzne

Proces haszowania składa się z dwóch kroków:

- transformacji (ang. folding) – pola nienumeryczne są transformowane na liczby całkowite
- haszowanie

Przykładowy algorytm transformacji

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

Kolizja

- Zasadniczym problemem w przypadku organizacji haszowej jest problem kolizji. Żadna funkcja haszowa nie gwarantuje, że różnym wartościom pola haszowego będą odpowiadały różne adresy szczylin
- **Kolizja** – gdy wartość funkcji haszowej dla danej wartości pola haszowego nowego rekordu odpowiada adresowi szczyliny, w której znajduje się już inny rekord
- **Rozwiązywanie kolizji** – proces znajdowania innej lokalizacji dla wprowadzanego rekordu

Metody rozwiązywania kolizji

- Metody rozwiązywania kolizji:
 - **adresowanie otwarte** (ang. open addressing) – następna wolna lokalizacja,
 - **łańcuchowanie** (ang. chaining) – wskaźniki do nowych lokalizacji,
 - **wielokrotne haszowanie** (ang. multiple hashing) – nowa funkcja haszowa.
- Każda metoda rozwiązywania kolizji wymaga własnych algorytmów wstawiania, wyszukiwania i usuwania rekordów

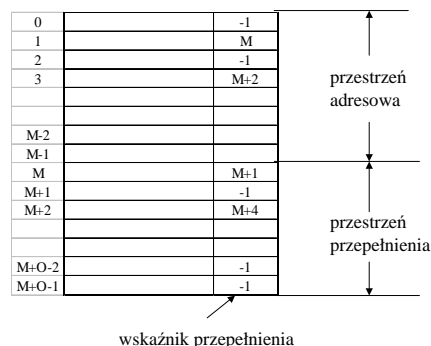
Metoda adresacji otwartej

```

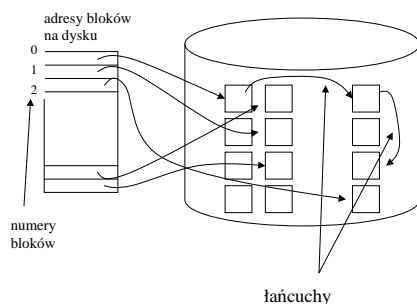
i := adres ;
if lokalizacja i jest zajęta then
  begin
    i := (i + 1) mod M;
    while (i ≠ adres) and lokalizacja i jest zajęta
      do i := (i + 1) mod M;
    if (i = adres)
      then wszystkie pozycje są zajęte;
    else nowy_adres := i;
  end;

```

Metoda łańcuchowania



Haszowanie zewnętrzne



Haszowanie zewnętrzne

- W haszowaniu zewnętrznym wartościami tablicy haszowej są adresy bloków dyskowych. Liczba bloków dyskowych tworzących plik haszowy jest stała i równa liczbie szczelin w tablicy haszowej
- Operacje:
 - przeglądanie pliku:** koszt wynosi $1.25N(D + RC)$. ponieważ wszystkie bloki muszą być odczytane
 - wyszukiwanie rekordu:** koszt wynosi $H + D + RC$
 - wyszukiwanie z przedziałem wartości:** koszt wynosi $1.25N(D + RC)$

Haszowanie zewnętrzne

- wstawianie rekordu:** koszt ten jest równy kosztowi wyszukiwania $+(C + D)$
- usuwanie rekordu:** koszt ten jest równy kosztowi wyszukiwania $+(C + D)$
- Cechą dobrej funkcji haszowej jest zapewnienie równomiernego rozkładu rekordów w obrębie przestrzeni adresowej tablicy haszowej. Zalecany rozmiar tablicy haszowej:

$$r/M \in (0,7 \div 0,9)$$
 gdzie r oznacza liczbę rekordów, natomiast M liczbę bloków

Dobra funkcja haszowa

- Dobra funkcja haszująca to taka, która odwzorowuje wartości kluczy, rozłożone nierównomiernie w obrębie dużej dziedziny, w przestrzeń adresową rekordów, o znacznie mniejszym rozmiarze w taki sposób, że wynikowe adresy są równomiernie rozłożone w obrębie przestrzeni adresowej
- Podstawowe kategorie funkcji haszujących:
 - haszowanie kongruencyjne** - $H(k) = k \bmod B$
 - potęgowanie** - podnosimy K do potęgi n i wybieramy $\log_2 B$ bitów jako adres strony
 - transformacja bazowa** - wartość K jest transformowana do systemu liczbowego o podstawie r ; część cyfr nowej reprezentacji jest wybierana jako adres

Jak dobrać parametry funkcji haszowej

- **Haszowanie kongruencyjne**

r – maksymalna liczba krotek relacji

bfr – liczba krotek w bloku

$$B = \lceil r/(bfr \cdot 0,8) \rceil$$

- **Problem:**

Założmy, że $B = x \cdot y$ i wartości kluczy są budowane następująco: $k_i = k_0 + i \cdot y$. Jeżeli tak, to:

$$H(k_i) = k_i \bmod B = k_0 \bmod x \cdot y + i \cdot y \bmod x \cdot y$$

$$H(k_i) = r_0 + i \bmod x$$

Estymacja liczby kolizji

Liczba kolizji jest zależna wprost proporcjonalnie od rozmiaru bloków i odwrotnie proporcjonalnie od stopnia wypełnienia bloków

Wypełnienie	Maksymalna liczba rekordów na stronę						
strony	1	2	3	5	10	20	40
0.5	0.500	0.177	0.087	0.031	0.005	0.000	0.000
0.6	0.750	0.293	0.158	0.066	0.015	0.002	0.000
0.7	1.167	0.494	0.286	0.136	0.042	0.010	0.001
0.8	2.000	0.903	0.554	0.289	0.110	0.036	0.009
0.85	2.832	1.316	0.827	0.449	0.185	0.069	0.022
0.9	4.495	2.146	1.377	0.777	0.345	0.144	0.055
0.92	5.740	2.768	1.792	1.024	0.467	0.203	0.083
0.95	9.444	4.631	0.035	1.769	0.837	0.386	0.171

Charakterystyka organizacji haszowej plików

- **Wady:**

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