Žilinská univerzita v Žiline

Fakulta riadenia a informatiky

Algoritmy a údajové štruktúry 2

Semestrálna práca č. 2

Marek Zaťko
5ZZT11

2018/19

Zvolená implementácia

Implementované bolo dynamické hashovanie s plnou funkcionalitou:

- **Všetky základné operácie**: find, insert, delete
- **Riešenie kolízií pomocou preplňujúceho súboru**
- **Manažovanie voľných blokov všetkých binárnych súborov**

Práca je implementovaná v jazyku Java, na GUI bola použitá platforma JavaFX.

Popis základných tried

Dynamic_hashing - trieda predstavujúca samotné dynamické hashovanie, pracuje s dvoma súbormi s priamým prístupom RandomAccessFile:

1. **Dátový súbor**: súbor obsahujúce dátové bloky
2. **Preplňujúci súbor**: súbor obsahujúci preplňujúce bloky.

Keďže pre potreby implementácie zadaní sú potrebné 2 inštancie dynamického hashovanie používame 4 súbory:

1. **byID.bin** - dátové bloky obsahujúce kľúčový atribút ID nehnuteľnosti a adresu v neutriedenom súbore obsahujúci všetky informácie o nehnutelnostiach.
2. **byID_over.bin** - preplňujúci súbor k súboru byID.bin
3. **bySCN.bin** - dátové bloky s kľúčovými atribútmi súpisné číslo nehnuteľnosti a názov katastrálneho územia a adresu v spomínanom neutriedenom súbore.
4. **bySCN_over.bin** - preplňujúci súbor k súboru bySCN.bin

Parametre konštruktora: počet relevantných bitov hashu, blokovací faktor dátových blokov a blokovací faktor preplňujúcich blokov a referencia na triedu, ktorá bude záznamom v danej inštancii dynamického hashovania.

DataBlock - trieda predstavuje jeden blok v binárnom súbore. Záznamy si drží v lineárnom zozname a informácie o platnosti týchto záznamov si drží v bitovom poli BitSet. **Vyhľadávanie konkrétneho záznamu je pre to rýchle pretože stačí prejsť len tie indexy, ktoré ma toto bitové pole nastavené na 1.** Mazanie je len vyhľadanie konkrétneho indexu v bitovom poli a následné nastavenie tohto bitu na 0.
Veľkosť tohto bitového pola v súbore je počet možných záznamov(blokovací faktor) v bloku zaokrúhlených na násobok 8, pretože pole ukladáme ako pole bajtov. **Napr. bitové pole pre 12 záznamov bude uložené ako 2 bajty.**

Štruktúra bloku v binárnom súbore:

Bitové pole platnosti záznamov (byte[])	Adresa nasledujúceho preplňujúceho bloku v zreťazení v preplňujúcom súbore (long)	Dĺžka zreťazenia preplňujúcich blokov (int)	**Záznamy**

FreeBlocksManager - manažér voľných blokov súboru s priamym prístupom. Pracuje s lineárnym zoznamom kde adresy voľných blokov sú usporiadané vzostupne. Ak sa dá tak skracuje binárny súbor a prideľuje voľné adresy novým blokom.

Dynamické hashovanie používa 2 inštancie tohto manažéra, jeden pre dátový súbor, a druhý pre preplňujúci.

UnsortedFIle - neutriedený binárny súbor s priamym prístupom. Súbor nie je blokovaný teda ku každému záznamu vieme pristúpiť priamo. Takisto využíva inštanciu FreeBlocksManager, kde ale veľkosť bloku je rovná veľkosti jedného záznamu. V práci je využitý na uchovanie všetkých informácií o nehnuteľnostiach, súbor: **nehnutelnosti.bin**

Record - interface, ktorý musí trieda ktorá bude záznamom v dynamickom hashovaní/neutriedenom súbore implementovať.

Nehnutelnost - predstavuje všetky informácie o nehnuteľnosti, implementuje Record, nehnutelnosti sú uložené v neutriedenom súbore nehnutelnosti.bin.

Nehnutelnost_byID - predstavuje pár ID a pozíciu(offset) v neutriedenom súbore, kde sa nachádzajú všetky údaje o nehnuteľnosti.

Nehnutelnost_byScN - predstavuje súpisné číslo, názov katastru a pozíciu(offset) v neutriedenom súbore, kde sa nachádzajú všetky údaje o nehnuteľnosti.

Počet prístupov do súboru pri operáciách find insert, delete

Jeden blokový prenos pozostáva z troch operácií:

1. Nastavenie pozície v binárnom súbore, operácia seek
2. Čítanie/zapísanie požadovaného počtu bajtov
3. Vytvorenie inštancie DataBlock z načítaného poľa bajtov

(v prípade čítania)

Operácia <u>Find(Record rec)</u>:

1. **Nájdenie externého vrchola v Trie-i podľa hashu**
2. **Načítanie dátového bloku z adresy, ktorú má vrchol**
3. **Ak blok obsahuje záznam, return nájdený**
4. **Ak blok má adresu ďalšieho preplňujúceho bloku v preplňujúcom súbore , tak ho načítaj a chod na 3. Ak túto adresu nemá, respektívne je nastavená na -1, tak hľadaný záznam sa v dynamickom hashovaní nenachádza.**

Teda počet blokových prenosov je v najhoršom prípade počet blokov v zreťazení.

V operačnej pamäti je teda načítaný súčasne len jeden blok.

<u>Zadanie 1</u> : Find v dynamickom hashovaní podľa súp.č a názvu -> získame offset, kde je záznam a prečítame záznam na danom offsete v neutriedenom súbore (1 prenos)

<u>Zadanie 2</u> : Find v dyn.hashovaní podľla ID, a zvyšok je rovnaký ako zadanie 1.

Operácia Insert(Record rec):

1. **Nájdenie externého vrchola v Trie-i podľa hashu.**

Ak adresa z vrchola je -1 tak vytvoríme nový blok a pridáme doňho záznam.

Spolu: 1 blokový prenos

Koniec

---

2. **Načítanie dátového bloku**

Ak počet záznamov v bloku < blokovací faktor dátových blokov tak pridáme záznam a zapíšeme blok

Spolu: 2 blokové prenosy

Koniec

Ak je hĺbka externého vrchola menšia ako maximálny počet relevantných bitov hashovacej funkcie a dátový blok nemá žiadny preplňujúci blok, tak skúsime rozdeliť blok podľa ďalších bitov hashovacej funkcie:

1. Ak sa podarí bloky rozdeliť tak zapíš novo vzniknuté bloky **+2 blokové prenosy**

Spolu: 3 blokové prenosy

Koniec

2. Ak sa bloky nepodarilo rozdeliť došlo ku kolízií:

2.1. Ak dátový blok nemá v zreťazení žiadny blok tak **vytvoríme nový preplňujúci blok v preplňujúcom súbore**, dátovému bloku zvýšíme dĺžku zreťazenia a nastavíme adresu ďalšieho bloku. Zapíšeme dátový blok a novovzniknutý preplňujúci blok **+ 2 blokové prenosy**

Spolu: 3 blokové prenosy

Koniec

2.2: Ak dátový blok má v zreťazení preplňujúce bloky, tak postupne načítavame tieto bloky a kontrolujeme počet platných prvkov v nich:

2.2.1: Ak nájdeme voľné miesto pridáme prvok a zapíšeme tento preplňujúci blok.

Spolu: 1 + počet načítaných blokov v zreťazení + 1 prenos

Koniec

2.2.2: Ak nenájdeme voľné miesto pridáme nový blok na koniec zreťazenia, aktualizujeme adresu nasledujúceho bloku predposlednému bloku a zvýšime dĺžku zreťazenia:

Spolu: 1+ počet blokov v zreťazení + 3 prenosy

Koniec

<u>Zadanie 3</u>: 1x insert do neusporiadaného súboru (1 prenos) + 1x insert do dyn. hashovania podľa ID + 1x insert do dyn. hashovania podľa súp.čísla a katastru

Operácia delete(Record rec):

1. **Nájdenie externého vrchola v Trie-i podľa hashu.**

Ak adresa z vrchola je -1 tak record určite v hashovaní neexistuje.

Koniec

---

2. **Načítanie dátového bloku**

---

3. **Postupne načítavame dátový blok a bloky v jeho zreťazení:**
   3.1. Ak nájdeme hľadaný prvok v dátovom bloku:
      3.1.1. Zmažeme prvok z bloku(nastavenie príslušného bitu na 0)
      3.1.2. Aktualizujeme počet valídnych prvkov vo vrchole

      3.1.3. Ak blok nemá zreťazenie a neostal prázdny, cyklické zlučovanie s bratským blokom (počet prenosov = počet načítaných bratov a zapísaní spojeného bloku)

      3.1.4. Ak blok nemá zreťazenie a ostal prázdny, aktualizuj adresu vo vrchole na -1 a pridaj adresu do voľných blokov a ak sa dá tak orež súbor

      3.1.5. Ak blok neostal prázdny a má zreťazenie, tak skúsime zlúčiť bloky v zreťazení, ak tým ušetríme aspoň jeden blok t.j ak počet voľných miest je väčší rovný ako blokovací faktor preplňujúcich blokov, tak začneme zlučovať:
      Ak je v súčasnom bloku nejaké voľné miesto tak skopíruj tento počet prvkov z ďalšieho bloku(načítaný) v zreťazení do súčasného bloku,
      zapíš súčasný blok a presuň sa na ďalší. Týmto spôsobom nám vznikne prázdny blok na konci zreťazenia.

3.2. Prvok nájdený v preplňujúcom bloku:

3.2.1. Ak blok neostal prázdny, tak postupujeme ako v 3.1.5

3.2.2. Ak blok ostal prázdny, aktualizujeme adresu nasledujúceho bloku predchádzajúcemu bloku a aktualizujeme dlžku zreťazenia. A adresu pridáme medzi voľné bloky

<u>Zadanie 4:</u> 1x Find v dynamickom hashovaní podľa súp.č a názvu -> získame offset v neutriedenom súbore

+ 1x Find z neutriedeného súboru, získame súp.číslo a kataster

+ 3x delete(2x hashovanie, 1x neutriedený súbor)

<u>Zadanie 5:</u> 1x Find v dyn.hashovaní podľa ID +
1x Zadanie 4 +
1x Zadanie 3