Algoritmos e Estruturas de Dados II

Prof. Ricardo J. G. B. Campello

Indexação de Arquivos II:

Índices Simples Grandes & Indexação Secundária

Adaptado e Estendido dos Originais de:

Leandro C. Cintra
Maria Cristina F. de Oliveira

Arquivo de Índice (Revisão)

- Exemplo Prático (Arquivo de Músicas)
  - Registros de tamanho variável com:
    - **ID Number:** Número de identificação
    - **Title**: Título
    - **Composer**: Compositor(es)
    - **Artist**: Artista(s)
    - **Label**: Rótulo (código da gravadora)
  - Chave primária:
    - Combinação de **Label** e **ID Number**

Arquivo de Índice (Revisão)

Record address	Label	ID number	Title	Composer(s)	Artist(s)
17	LON	2312	Romeo and Juliet	Prokofiev	Maazel
62	RCA	2626	Quartet in C Sharp Minor	Beethoven	Julliard
117	WAR	23699	Touchstone	Corea	Corea
152	ANG	3795	Symphony No. 9	Beethoven	Giulini
196	COL	38358	Nebraska	Springsteen	Springsteen
241	DG	18807	Symphony No. 9	Beethoven	Karajan
285	MER	75016	Coq d'Or Suite	Rimsky-Korsakov	Leinsdorf
338	COL	31809	Symphony No. 9	Dvorak	Bernstein
382	DG	139201	Violin Concerto	Beethoven	Ferras
427	FF	245	Good News	Sweet Honey in the Rock	Sweet Honey in the Rock

Figure 7.2 Contents of sample recording file.

Arquivo de Índice (Revisão)

Index

Key	*Reference field*
ANG3795	152
COL31809	338
COL38358	196
DG139201	382
DG18807	241
FF245	427
LON2312	17
MER75016	285
RCA2626	62
WAR23699	117

Recording file

Address of record	*Actual data record*
17	LON | 2312 | Romeo and Juliet | Prokofiev | ...
62	RCA | 2626| Quartet in C Sharp Minor | Beethoven | ...
117	WAR | 23699 | Touchstone | Corea | ...
152	ANG | 3795 | Symphony No. 9 | Beethoven | ...
196	COL | 38358 | Nebraska | Springsteen | ...
241	DG | 18807 | Symphony No. 9 | Beethoven | ...
285	MER | 75016 | Coq d'Or Suite | Rimsky-Korsakov | ...
338	COL | 31809 | Symphony No. 9 | Dvorak | ...
382	DG | 139201 | Violin Concerto | Beethoven | ...
427	FF | 245 | Good News |Sweet Honey in the Rock | ...

Figure 7.3 Index of the sample recording file.

Arquivos de Índice Grandes

- Se o índice não cabe na memória primária, o acesso e manutenção precisam ser feitos em memória secundária
  - Nada muda para o arquivo principal, que é manipulado em memória secundária sempre
- **Busca**
  - Busca seqüencial é O(n) acessos, mesmo com blocagem
  - BB é O(log n) acessos, mas não se beneficia de blocagem
    - pode demandar um acesso para cada registro verificado

5

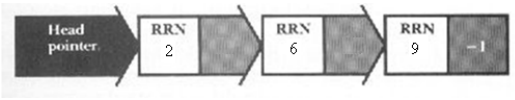
Arquivos de Índice Grandes

- Se o índice não cabe na memória primária, o acesso e manutenção precisam ser feitos em memória secundária
- **Remoção**
  - Alternativa 1: Deslocar todos os registros subseqüentes no arquivo de índice para preencher espaço do registro removido
    - otimiza espaço, mas a um custo computacional altíssimo...
  - Alternativa 2: Colocar um marcador e encadear o registro removido em uma lista de registros de índice disponíveis
    - análogo ao que é feito para o arquivo principal

6

Arquivos de Índice Grandes

- **Remoção**
  - Alternativa 2 (Exemplo):



head.first_avail = 2

RRN	*Key*	*Reference field*
0	ANG3795	152
1	COL31809	338
2	*|6|COL38358	196
3	DG139201	382
4	DG18807	241
5	FF245	427
6	*|9|LON2312	17
7	MER75016	285
8	RCA2626	62
9	*|-1|WAR23699	117

Index

Arquivos de Índice Grandes

- **Remoção**
  - Alternativa 2 (limitação):
    - inserção deverá respeitar ordem da chave para permitir BB ...
    - pode não valer a pena manter e percorrer a lista de disponíveis com baixa possibilidade de sucesso ...

head.first_avail = 2

RRN	*Key*	*Reference field*
0	ANG3795	152
1	COL31809	338
2	*|6|COL38358	196
3	DG139201	382
4	DG18807	241
5	FF245	427
6	*|9|LON2312	17
7	MER75016	285
8	RCA2626	62
9	*|-1|WAR23699	117

Index

Arquivos de Índice Grandes

- **Remoção**
  - Alternativa 3:
    - Apenas marcar os registros como disponíveis (sem lista)

RRN	*Key*	*Reference field*
0	ANG3795	152
1	COL31809	338
2	* COL38358	196
3	DG139201	382
4	DG18807	241
5	FF245	427
6	* LON2312	17
7	MER75016	285
8	RCA2626	62
9	* WAR23699	117

Index

Arquivos de Índice Grandes

- Se o índice não cabe na memória primária, o acesso e manutenção precisam ser feitos em memória secundária
- **Inserção** (alternativa 3 de remoção)
  - Para permitir BB, chave inserida deve respeitar ordem do índice
    - Busca-se pela localização onde a chave deveria ser inserida (BB)
    - Se localização corresponde a um slot disponível, tudo resolvido
    - Caso contrário, é necessário deslocar todos os registros de índice subseqüentes até o próximo slot vago ou EOF

10

Arquivos de Índice Grandes

- Se o índice não cabe na memória primária, o acesso e manutenção precisam ser feitos em memória secundária
- **Atualização**
  - Se atualização muda o valor da chave:
    - trata-se como uma remoção do reg. de índice antigo seguida de uma inserção do reg. de índice atualizado
  - Se atualização não muda o valor da chave:
    - se tamanho do registro não aumenta, nada muda no índice
    - caso contrário, muda-se apenas o byte offset no índice

11

Arquivos de Índice Grandes

- Desempenho das operações em arquivos de índices simples que não cabem em RAM só pode ser melhorado com abordagens de indexação mais sofisticadas:
  - Hashing Externo
    - Máximo desempenho para acesso direto
  - Árvores
    - Bom compromisso entre desempenho, manutenibilidade e possibilidade de acesso seqüencial ordenado por chaves

12

Indexação Secundária

- O que fazer quando a chave primária não é o alvo da consulta?
  - Por exemplo, enquanto CPF é uma chave muito usual, o que dizer do código do nosso arquivo de músicas?
    - Como saber que se deve procurar por **COL38358** quando se deseja a ficha musical de "Nebraska", de Bruce Springsteen ???

13

Indexação Secundária

- Muitas vezes, o acesso a registros não se faz por chave primária, mas por chaves secundárias
- Como localizar o registro, se nosso índice é construído em função da chave primária?
- **Solução**:
  - cria-se um outro índice que relaciona uma chave secundária à chave primária (late binding)
  - usa-se então o índice da chave primária para localizar o registro

14

Indexação Secundária

- Exemplo Prático (Arquivo de Músicas):

Composer index

Secondary key	*Primary key*
BEETHOVEN	ANG3795
BEETHOVEN	DG139201
BEETHOVEN	DG18807
BEETHOVEN	RCA2626
COREA	WAR23699
DVORAK	COL31809
PROKOFIEV	LON2312
RIMSKY-KORSAKOV	MER75016
SPRINGSTEEN	COL38358
SWEET HONEY IN THE R	FF245

Title index

Secondary key	*Primary key*
COQ D'OR SUITE	MER75016
GOOD NEWS	FF245
NEBRASKA	COL38358
QUARTET IN C SHARP M	RCA2626
ROMEO AND JULIET	LON2312
SYMPHONY NO. 9	ANG3795
SYMPHONY NO. 9	COL31809
SYMPHONY NO. 9	DG18807
TOUCHSTONE	WAR23699
VIOLIN CONCERTO	DG139201

15

Indexação Secundária

- Índices permitem muito mais que melhorar o tempo de localização de um registro
- Múltiplos índices secundários:
  - permitem manter diferentes visões dos registros em um mesmo arquivo de dados
  - permitem combinar chaves associadas e fazer consultas que combinam visões particulares

16

Indexação Secundária

- Diferença importante entre os índices dos tipos primário e secundário:
  - Nos secundários, podem ocorrer múltiplos registros com chaves iguais
  - Chaves duplicadas devem ser mantidas agrupadas e ordenadas internamente ao grupo segundo a chave primária
    - Permite consultas eficientes envolvendo combinações de chaves secundárias...

17

Operações Básicas

- **Remoção:**
  - Implica em remover o registro do arquivo de dados e de todos os arquivos de índices
  - Buscar o registro e eventualmente gerenciar os espaços vagos resultantes em múltiplos arquivos de índices pode ser custoso se não couberem em RAM
  - **Alternativa:** atualizar apenas o índice primário, sem eliminar as entradas correspondentes nos índices secundários

18

Operações Básicas

- **Remoção** (alternativa)**:**
  - Atualizar apenas o índice primário, sem eliminar as entradas correspondentes nos índices secundários
    - É mais simples e menos sujeito a inconsistências
    - A busca irá apenas ser mal sucedida ao procurar, a partir de uma referência não atualizada no arquivo de índice secundário, por uma chave primária que não mais existe
      - Nesse momento, é possível eliminar o registro do índice secundário
      - Porém, existe um custo computacional extra associado
        - Busca por chave inexistente no índice primário

19

Operações Básicas

- **Inserção:**
  - Quando um novo registro é inserido no arquivo, devem ser inseridas as entradas correspondentes no índice primário e nos índices secundários
    - entradas devem ser inseridas respeitando a ordenação
    - se os arquivos de índices não couberem em RAM, pode ser muito custoso
      - especialmente quando a remoção alternativa é adotada, o que implica necessariamente o deslocamento de todos os registros subseqüentes à posição de inserção até o final do arquivo.

20

Operações Básicas

- **Atualização** (3 situações):
  - Situação 1: Alterou uma chave secundária
    - índice secundário para esta chave precisa ser reordenado

21

Operações Básicas

- **Atualização** (3 situações):
  - Situação 2: Alterou a chave primária
    - índice primário precisa ser reordenado
    - índices secundários precisam ser varridos e as entradas contendo a chave primária alterada devem ser atualizadas
      - se houver chaves secundárias duplicadas, pode ser necessário reordená-las localmente pela chave primária

22

Operações Básicas

- **Atualização** (3 situações):
  - Situação 3: Alterou apenas outros campos
    - não afeta nenhum dos índices
    - no máximo é preciso atualizar o valor do byte offset no respectivo registro do índice primário
      - Porque...?

23

Exercícios

- Capítulo 7 (Folk & Zoellick, 1987)
- Lista de Exercícios (CoTeia)
  - **Nota.** A lista faz referências à 2ª edição do livro de Folk & Zoellic. Nesse caso, o capítulo de indexação é o Capítulo 6
    - FOLK, M. & ZOELLICK, B., *File Structures*, 2nd Edition, Addison-Wesley, 1992.

24

Bibliografia

- **M. J. Folk and B. Zoellick, *File Structures: A Conceptual Toolkit*, Addison Wesley, 1987.**

25