

ICMC-USP
SCC0603 – Algoritmos e Estruturas de Dados II
Prof. Rosane Minghim – 1º Semestre 2012

Trabalho 3 – Índices

Esse trabalho consiste na criação e atualização de um índice para facilitar a manipulação de registros de um arquivo muito grande para ser mantido completamente na memória primária. Utilizaremos uma versão simples de um conceito muito utilizado na prática (de forma mais complexa) em sistemas de gerenciamento de bancos de dados.

A entrada consistirá em um conjunto de referências bibliográficas (o nosso banco de dados) seguido de um conjunto de operações a serem realizadas nesses dados: adição, remoção e modificação de registros. Como saída, você deverá imprimir o banco de dados após a aplicação das operações indicadas.

1 Detalhes da entrada

Abaixo segue um exemplo de entrada e a explicação de cada parte:

```
2
Leung, Joseph Y.-T./A new algorithm for scheduling periodic, real-time
  tasks/Algorithmica/4/2/1989/209-219/0178-4617
Jansen, Klaus/Scheduling Malleable Parallel Tasks An Asymptotic Fully
  Polynomial Time Approximation Scheme/Algorithmica/39/1/2004/59-81/
  0178-4617
3
ADD/Anderson, Richard J. and Miller, Gary L./Deterministic Parallel List
  Ranking/Algorithmica/6/6/1991/859-868/0178-4617
MODIFY/0178-4617/4/2/209-219/title/On Independent Vertex Sets in Subclasses
  of Apple-Free Graphs
REMOVE/0178-4617/39/1/59-81
```

A primeira linha fornece o número N de referências que formam o banco de dados. As N linhas seguintes são as referências; cada referência possui vários campos, que estão no formato:

```
author/title/journal/volume/number/year/pages/issn
```

Observe que não há espaços em branco ou quaisquer outros símbolos separando cada campo, apenas o ”/”. Além disso, não há quebras nas linhas (aqui no enunciado as linhas

estão quebradas para que caibam na página). O ISSN é um número que identifica de forma única um *journal* no mundo.

A primeira linha após as N referências fornece o número M de operações a serem realizadas no banco de dados. As M linhas seguintes são as operações, que podem ser de três tipos:

1. **Adicionar registro:** A operação de adição de registro é identificada por **ADD**, seguida de uma referência completa no mesmo padrão mostrado acima.
2. **Modificar registro:** A operação de modificação de registro é identificada por **MODIFY**, seguida de:
 - (a) Uma sequência "ISSN/Volume/Número/Páginas" que identifica (de forma única) o registro a ser modificado;
 - (b) O nome do campo a ser modificado no registro;
 - (c) O novo valor do campo.
3. **Remover registro:** A operação de remoção de registro é identificada por **REMOVE**, seguida de uma sequência "ISSN/Volume/Número/Páginas" que identifica (de forma única) o registro a ser removido.

2 Detalhes da saída esperada

Você deverá imprimir o banco de dados de referências de volta na saída padrão após a realização de todas as operações indicadas. A ordem dos registros deve obedecer a ordem crescente da chave, e o formato deve ser exatamente o mesmo da entrada. A chave é a sequência única descrita nas operações "REMOVE" e "MODIFY".

Com a entrada fornecida acima, o resultado seria:

```
Leung, Joseph Y.-T./On Independent Vertex Sets in Subclasses  
of Apple-Free Graphs/Algorithmica/4/2/1989/209-219/0178-4617  
Anderson, Richard J. and Miller, Gary L./Deterministic Parallel List  
Ranking/Algorithmica/6/6/1991/859-868/0178-4617
```

3 Implementação

Os recursos disponíveis (memória e processamento) para o trabalho serão extremamente limitados; ou seja, não será possível carregar todos os dados na memória para realizar as operações. Você deverá implementar um índice na memória que contenha apenas uma chave primária de cada registro e sua localização no arquivo. Assim, pouca memória será utilizada e a busca pelo registro no arquivo pode ser feita quase instantaneamente.

4 Observações

- A entrada foi formatada especificamente para facilitar a leitura com a função `strtok`. Exemplo:

```
fgets(line, MAX_LINE_CHARS, stdin);
author = strtok(line, "/");
title = strtok(NULL, "/");
...
```

- Para gravar os dados em um arquivo, utilize o diretório atual do programa; por exemplo:

```
FILE *f = fopen("banco.dat", "w+");
```

- Cada `MODIFY` modifica um campo por vez.
- A adição de registros (`ADD`) deve ser sempre no final do arquivo.
- Mantenha o índice na memória ordenado e utilize busca binária para achar os registros a serem manipulados.