

ALGORITMOS E ESTRUTURAS DE DADOS II

Trabalho 2 – Primeiro Semestre – 2010

Profa. Rosane Minghim

Descrição

Um pesquisador deseja manter um cadastro de suas referências bibliográficas (artigos e livros) em disco, e acabou de contratá-lo para resolver o problema. Para cada referência, o pesquisador deseja manter os seguintes atributos:

- Código da referência (composição das três primeiras letras do nome do primeiro autor e ano da publicação, por ex. SHI90 – esse campo é chave).
- Título (string com o título).
- Autor (sobrenome e iniciais do primeiro autor, por ex.: Schimman, D.E.).
- Ano de publicação (por ex.: 1990).
- Veículo (string longa com o nome do congresso ou periódico, e outros dados adicionais, como páginas, volume e número, local, etc.).

Você deverá desenvolver um sistema que permita ao pesquisador:

- Inserir uma nova referência, atualizando um índice.
- Remover uma referência a partir da chave. O registro deverá ser localizado acessando o índice. A remoção deve colocar o caractere # (sustenido) na primeira posição do registro removido. O espaço do registro removido não deverá ser reutilizado para novas inserções, as quais deverão ser colocadas sempre no fim do arquivo (supõe-se, assim, que um processo de compactação eliminaria os espaços disponíveis em outro processo, o qual não é necessário implementar).
- Buscar uma referência a partir da chave. A busca deve ser realizada acessando o índice.

Ao final da execução, dois arquivos deverão ter sido criados:

- O arquivo de índice deve ser binário. A chave deve ser declarada como um vetor de 6 caracteres (um a mais para o '\0'). O endereço (*offset*) deve ser do tipo *long* de 4 bytes. Como os campos do índice possuem tamanho fixo, tudo deve ser armazenado sem caracteres delimitadores nem quebras de linha. **Obrigatoriamente o arquivo de índice deverá ser chamado de "index.dat"**.
- O arquivo de dados deve ser ASCII, e organizado em registros de tamanho fixo de 256 bytes, com os campos título, autor e veículo de tamanho variável, e os demais de tamanho fixo. Os campos no registro devem ser separados pelo caractere delimitador @ (arroba). Após o último caractere @, que indica o fim do registro, preencher o restante do espaço com o caractere # (sustenido). Por exemplo:

key01 @título1 @autor1 @ano1 @veículo1 @##...##key02 @título2 @autor2 @ano2 @veículo2 @#...

Obrigatoriamente o arquivo de dados deverá ser chamado de "data.dat".

Nenhum dos dois arquivos deverá ter cabeçalho.

Comandos Permitidos

Os comandos permitidos são representados por duas letras maiúsculas. Toda linha de entrada obrigatoriamente inicia com um comando. Só serão fornecidos como entrada comandos aqui especificados (não há necessidade de tratar comandos inválidos, pois só serão fornecidos comandos válidos). Os únicos comandos permitidos como entrada são apresentados a seguir. O símbolo ¶ denota um único espaço em branco.

Inserir referência

```
IR ¶ <chave> ¶ <título> ¶ <autor> ¶ <ano> ¶ <veículo>
```

Inserir uma nova referência contendo todos os campos, como especificado acima.

Remover referência

```
RR ¶ <chave>
```

Remover uma referência a partir da sua chave.

Buscar referência

```
BR ¶ <chave>
```

Buscar uma referência a partir da sua chave e imprimir o registro da seguinte forma:

```
<chave> ¶ <título> ¶ <autor> ¶ <ano> ¶ <veículo>
```

Finalizar a execução

```
FM
```

Finalizar a execução do programa, gravando o arquivo de índice no disco e fechando-o, juntamente com o arquivo de dados.

Entrega e Observações Importantes

- O terceiro trabalho consistirá em efetuar a troca do índice por outra estrutura (árvore-B). Para facilitar a realização do próximo trabalho, é muito importante que o arquivo de dados seja manipulado de forma completamente independente do índice. Tente projetar seu programa de modo que, no terceiro trabalho, não seja necessário alterar nenhuma linha do programa principal nem das funções de manipulação do arquivo de dados.
- Implemente o trabalho usando a linguagem C padrão ANSI ou em C++.
- Vazamento de memória, referência a valores de variáveis não inicializadas e outros defeitos serão levados em conta na avaliação.
- Os trabalhos deverão ser feitos **individualmente**.
- Serão anulados aqueles trabalhos nos quais forem detectados quaisquer tipos de cópia ou plágio, não importa a origem.
- A data limite para entrega é 13/06.
- Os trabalhos deverão ser entregues pelo SQTPM em arquivo único (".c" ou ".cpp").