

Trabalho 2 – Arquivos e Índices

Data de Entrega: 08/06/2014

Neste trabalho o objetivo é implementar um sistema de manipulação de um arquivo de dados com índices associados. O arquivo de dados é mantido em disco, assim como os arquivos de índice (um índice primário, e dois índices secundários). Por meio do sistema a ser desenvolvido, o arquivo de dados admite operações de inserção e remoção, as quais devem ser refletidas nos arquivos de índice. Entretanto, a manipulação dos índices é feita com os mesmos mantidos em memória principal.

Arquivo de dados

O arquivo de dados contém registros sobre pessoas vinculadas à USP de São Carlos. Sobre esse arquivo serão realizadas operações de inserção, remoção, impressão e modificação dos registros. Cada registro consiste de 5 campos, e os registros têm tamanho fixo, de 80 caracteres, com os campos delimitados por um caracter especial ('|'). A forma canônica das chaves é tudo em minúsculo.

- 1.NUSP
- 2.Nome
- 3.Unidade(eesc /icmc /ifsc/iqsc/iau/iea/cdce)
- 4.Vínculo(aluno/funcionário/professor)
- 5.Ano de Ingresso

Por exemplo:

6054844 mary smith icmc aluno 2012.....

Sistema

O campo NUSP é chave primária, os campos Unidade e Vínculo são chaves secundárias. Os valores das chaves sempre devem ser fornecidos como entrada para que o registro seja criado. As seguintes operações são admitidas pelo sistema:

INDICES <arquivo> (cria os índices a partir de um arquivo de dados disponível)

ADD nusp='xxxx' unidade='xxxx' vínculo='xxxx' nome='xxxx' ingresso='xxxx'
enter

MODIFY nusp='xxxx', tag='value' *enter*

REMOVE nusp='xxxx' *enter*

PRINT nusp='xxxx', tag1, tag2, tag3 *enter* (ou * para imprimir todos os tags)

PRINT vinculo='xxxx', tag1, tag2, tag3 *enter*

PRINT unidade='xxxx', tag1, tag2, tag3 *enter*

Em que *tag* pode ser **nome**, **ano de ingresso**, **vinculo**, **unidade**. Cada comando é encerrado por um <enter>, as entradas não são *case sensitive*, ou seja um comando não é sensível a maiúsculas e minúsculas.

As operações ADD e MODIFY devem imprimir em um arquivo o resultado da operação, se bem sucedida, caso contrário a palavra ERROR seguida da operação e uma mensagem indicativa do erro. Exemplos:

ERROR: ADD nusp='xxxx', tag='value' – nro usp já existe.

ERROR: MODIFY nusp='xxxx', tag='value' - nro usp não existe.

ERROR: REMOVE Nusp='xxxx', tag='value' – nro usp não existe.

ERROR: PRINT NUSP='xxxx', tag1, tag2, tag3 – nro usp não existe.

ERROR: PRINT vinculo='xxxx', tag1, tag2, tag3 – vínculo não existe.

ERROR: PRINT unidade='xxxx', tag1, tag2, tag3 – unidade não existe.

O sistema recebe uma sequência de comandos sem fazer nada, até o usuário entrar o comando EXECUTE, o qual provoca a execução da sequência de operações. Em seguida, aguarda novos comandos. Quando o usuário digitar EXIT, o sistema termina a execução. A saída é o arquivo log.

Exemplo de entrada/saída

Entrada:

```
INDICES dados.txt
ADD NUSP='4524879', unidade='ICMC', vinculo='professor', nome='Joao Silva'
ano='1990' enter
ADD NUSP='8684759', unidade='EESC', vinculo='aluno', nome='Mariana Santos'
ano='2012' enter
ADD NUSP='7584987', unidade='IQSC', vinculo='aluno', nome='Roberto Oliveira'
ano='2011' enter
ADD NUSP='73648796', unidade='IFSC', vinculo='aluno', nome='Valeria Pereira'
ano='2010' enter
EXECUTE

MODIFY nusp='8684759', unidade='IFSC' enter
REMOVE nusp='4325198' enter
REMOVE nusp='7584987' enter
PRINT nusp='7584987', * enter
PRINT nusp='8684759', * enter
PRINT vinculo='aluno', NUSP, nome enter
PRINT unidade='icmc', nome, ano enter
EXECUTE

EXIT
```

Saída:

```
índices carregados.
4524879|joao silva|icmc|professor|1990
8684759|mariana santos|eesc|aluno|2012
7584987|roberto oliveira|iqlsc|aluno|2011
73648796|valeria pereira|ifsc|aluno|2010
8684759|mariana santos|ifsc|aluno|2012 modificado com sucesso.
ERROR: REMOVE nusp='4325198'– nro usp não existe.
7584987|roberto oliveira|iqlsc|aluno|2011 removido com sucesso.
ERROR: PRINT nusp='7584987'– nro usp não existe.
8684759|mariana santos|ifsc|aluno|2012
8684759|mariana santos
73648796|valeria pereira
joao silva|1990
```

Implementação

Assume-se que os índices podem ser mantidos em memória principal, mas o arquivo de dados não. Os índices devem ser atualizados em disco sempre que for executada uma sequência de operações (i.e., após cada ocorrência de 'EXECUTE'). Os índices secundários devem referenciar as chaves primárias, apenas com marcação dos registros removidos. A busca por chaves nos índices secundários pode ser sequencial, mas no índice primário deve ser feita busca binária.

Outras Informações Importantes

- ❖ O trabalho deve ser feito em duplas (que não deve ser a mesma dos trabalhos anteriores), sendo que ambos os membros devem submetê-lo através do [Sistema de Submissão de Programas](#) (SSP) na disciplina **SCC603 – Algoritmos e Estruturas de Dados 2 (2014/1) Turma A**.
- ❖ Todas as submissões são checadas para evitar cópia/plágio/etc. Portanto, evite problemas e implemente o seu próprio código.
- ❖ Comente o seu código com uma explicação rápida do que cada função, método ou trecho importante de código faz (ou deveria fazer). Os comentários serão checados e considerados na avaliação.
- ❖ Mantenha a modularização entre código e dados, ou seja, divida o código explicitamente entre estruturas de dados e algoritmos de manipulação. A modularização será checada e considerada na avaliação.
- ❖ Apresentação de um relatório: Deve ser entregue um relatório com as informações necessárias e descrições das funcionalidades do seu sistema. O relatório será checado e considerado na avaliação.
 - A estrutura básica do seu relatório deve ser a seguinte:
 - Capa: Código e nome da disciplina, nomes completos e números USP dos integrantes do grupo.
 - Conteúdo:
 - Introdução: apresentação geral do trabalho e organização do documento.
 - Sistema: descrição do sistema, estruturas de dados utilizadas e

funcionalidades relevantes.

- Dificuldades apresentadas: relatar brevemente as principais dificuldades encontradas durante a implementação e as decisões tomadas para superá-las.