Projeto 3: Indexação – Árvore B

O objetivo do projeto é gerenciar um sistema de arquivos que gerência a locação de veículos de uma dada empresa. O sistema armazena as seguintes informações:

- Código do Cliente (CodCli)
- Código do Veículo (CodVei)
- Nome do Cliente
- Nome do Veículo
- Número de Dias

A chave primária é composta pela composição "CodCli+CodVei". Para simplificar, considere que um cliente pode alugar uma única vez um determinado carro (i.e., CodCli+CodVei será sempre único). O arquivo a ser criado deve ser **binário** e de registros e campos de tamanho variável-variável ou fixo-fixo.

Código do Cliente	Código do Veículo	Nome do Cliente	Nome do Veículo	Número de Dias
11 caracteres (fixo)	7 caracteres (fixo)	50 caracteres (máximo)	50 caracteres (máximo)	int (fixo ou caracteres)

As seguintes funcionalidades deverão estar disponíveis:

1. Inserção
2. Listar os dados de todos os clientes [percurso ordenado na árvore B]
3. Pesquisa por chave primária, i.e., "CodCli+CodVel" [índice primário - árvore B]
4. Carrega Arquivos (dependente da implementação)

1 - Inserção

Insere o registro no final do arquivo principal. Deve-se atualizar o arquivo de índice.

Os dados a serem inseridos devem ser recuperados de um arquivo a ser fornecido no momento da execução do programa (vide funcionalidade 4).

Este índice deve utilizar uma estrutura de Árvore-B (vide funcionalidade 3).

Portanto, a cada nova inserção as seguintes mensagens deverão ser mostradas, onde <X> é o valor da chave promovida (note que mais de uma pode aparecer):

- *"Divisão de nó"* [deve ser impressa sempre que um nó for dividido]
- *"Chave promovida: <X>"* [deve ser impressa sempre que uma chave for promovida]
- *"Chave inserida com sucesso: <X>"* [deve ser impressa ao final da inserção indicando sucesso da operação]
- *"Chave duplicada: <X>"* [deve ser impressa ao final da inserção e indica que a operação de inserção não foi realizada]

Observação: antes de inserir um registro no arquivo principal certifique-se de que a chave não existe no índice. Na avaliação do projeto será testado a inserção de chaves repetidas!

Exemplo de Inserção
<input> 11111111111ABC1234
<output> Chave inserida com sucesso: 11111111111ABC1234
<input> 33333333333CDE9874
<output> Chave inserida com sucesso: 33333333333CDE9874
<input> 22222222222ABC1234
<output> Chave inserida com sucesso: 22222222222ABC1234
<input> 44444444444ERT4561
<output> Divisão de nó
<output> Chave promovida: 33333333333CDE9874
<output> Chave inserida com sucesso: 44444444444ERT4561
<input> 33333333333CDE9874
<output> Chave duplicada: 33333333333CDE9874

2 - Listar os dados de todos os clientes

Nessa opção o índice baseado em árvore-B deverá ser percorrido em-ordem e a cada "CodCli+CodVel" encontrado listar os dados associados ao mesmo. Desse modo, essa opção deverá imprimir os dados de todos os clientes cadastrados por ordem de "CodCli+CodVel".

3 - Pesquisa por chave primária

Para essa funcionalidade deve-se gerenciar um arquivo de índice que contenha a lista das chaves primárias, i.e., "CodCli+CodVel", presentes no arquivo de dados junto com o deslocamento (byte offset) necessário para acessar o registro de cada chave presente no arquivo principal.

Este índice deve utilizar uma estrutura de Árvore-B. Para tanto, a busca deve ser feita na árvore-B. Além disso, as seguintes mensagens deverão ser exibidas em relação à busca na árvore:

- *"Chave encontrada: <X> | pag: <Y> | pos: <Z>"* [indica que a Chave <X> foi encontrada e encontra-se na página <Y>, na posição <Z> da página. Após a exibição dessa mensagem, os dados referentes ao cliente deverão ser recuperados do arquivo principal e apresentados na saída]
- *"Chave não encontrada: <X>"* [indica que a Chave <X> não está presente na árvore-B e, consequente, no arquivo principal.

Exemplo de Inserção
<input> 11111111111ABC1234
<output> Chave encontrada: 11111111111ABC1234 | pag: 0 | pos: 0
<output> *<dados do registro>*
<input> 33333333333CDE9874
<output> Chave não encontrada: 33333333333CDE9874

Assim, uma consulta deve primeiramente procurar a chave desejada no arquivo de índice (usado a abordagem de Árvore B) e depois acessar diretamente o registro desejado no arquivo de dados. Os dados relacionados ao "CodCli+CodVel" pesquisado devem ser exibidos.

Os dados a serem pesquisados devem ser recuperados de um arquivo a ser fornecido no momento da execução do programa (vide funcionalidade 4).

4 - Carrega Arquivos

A fim de facilitar os testes e avaliação do projeto, serão fornecidos dois arquivos:

a) "insere.bin"
b) "busca.bin"

O arquivo (a) conterá os dados a serem inseridos durante os testes. Não necessariamente todos os dados serão inseridos, isto é, está funcionalidade deve perguntar ao usuário qual registro deve ser inserido. Para tanto, uma sugestão é carregar o arquivo em memória (um vetor de struct, por exemplo) e ir acessando cada posição conforme as inserções vão ocorrendo.

O arquivo (b) conterá uma lista de chaves primarias, "CodCli+CodVei", a serem utilizados durante a pesquisa por chave primária (funcionalidade 3). A ideia é a mesma já descrita, ou seja, carregar o arquivo em memória (um vetor de struct, por exemplo) e ir acessando cada posição conforme as remoções vão ocorrendo.

Nesses arquivos os registros seguem uma organização de registro e campo de tamanho fixo. Ver códigos fonte fornecidos.

Observações gerais:

(1) Todos os arquivos (principal e índice) deverão ser manipulados em memória secundaria

(2) A inserção de um registro requer a manipulação de 2 arquivos (principal e índice).

(3) Não criar os arquivos toda vez que o programa for aberto (fazer verificação). Isto é, o programa pode ser encerrado, e ao recomeçar deve continuar com o arquivo do estado que parou!

(4) O programa deve realizar as operações sobre uma árvore-B de ordem 4 (ou seja, no máximo 3 chaves). Para padronizar, sempre promover, quando houver overflow, a chave de índice 1, começando em zero. Exemplo:

111…	222…	333…

Inserção de 444… causa overflow

111…	222…	333…	444…

Promover 222… (índice 1)

Para auxiliar o desenvolvimento do trabalho é fornecido um código que insere chaves em uma árvore-B de ordem 5. Vocês devem utilizar esse código como base. Entretanto, algumas alterações serão

necessárias para que o mesmo funcione corretamente. Vocês deverão estudar e entender o código para que consigam fazer as alterações necessárias. No caso do procedimento de pesquisa básica, tome como base o pseudocódigo discutido em sala de aula. Em relação ao procedimento de percurso em-ordem, o mesmo deverá ser desenvolvido.

(5) Criar um pequeno menu para acessar cada uma das funcionalidades (1, 2 e 3, sendo que a 4 pode ser ativada ao iniciar o programa). Note que as funcionalidades podem ser executadas de forma aleatório de acordo com a necessidade do usuário. Por exemplo, 3 inserções → 1 busca → 1 inserção → fechar/abrir programa → 2 inserções → 1 busca.

(6) A avaliação terá uma dinâmica como o seguinte exemplo:

1. execute o programa
2. insira um registro – 3 (índice do arquivo insere.bin)
3. insira um registro – 5 (índice do arquivo insere.bin)
4. insira um registro – 1 (índice do arquivo insere.bin)
5. feche o programa
6. abra os arquivos no editor hexadecimal
7. execute o programa novamente
8. pesquisa um registro – 2 (índice do arquivo busca.bin)
9. insira um registro – 2 (índice do arquivo insere.bin)
10. abra os arquivos no editor hexadecimal
11. execute o programa novamente
12. insira um registro – 4 (índice do arquivo insere.bin)
13. pesquisa um registro – 3 (índice do arquivo busca.bin)
14. listar todos os registros em ordem
15. feche o programa
16. abra o arquivo no editor hexadecimal
17. …