

## 2º Trabalho Prático

O programa referente ao 2º trabalho prático da disciplina Organização e Recuperação de Dados deverá ser desenvolvido na linguagem C por grupos de até duas pessoas. O código-fonte do trabalho deverá estar devidamente identificado com os nomes das pessoas do grupo e ser entregue exclusivamente via *moodle*.

### Especificação

O objetivo deste trabalho é criar um programa que, a partir do arquivo de registros de filmes criado no 1º trabalho, construa uma **árvore-B de ordem 5** (máximo de 4 chaves por página) que servirá de índice primário para o arquivo de registros.

Para facilitar o desenvolvimento do trabalho, o arquivo com os registros de filmes (*dados.dat*) será disponibilizado no *moodle* já no formato correto. Dessa forma, não será necessária a importação do arquivo *filmes.txt*. O arquivo *dados.dat* disponibilizado é iniciado por um cabeçalho de quatro bytes (**int**), assim como no 1º trabalho. Porém, em vez de armazenar a cabeça da LED, o cabeçalho armazena a quantidade de registros existente no arquivo *dados.dat*. O programa deverá ser capaz de lidar com arquivos contendo qualquer quantidade de filmes, desde que se respeite o formato do arquivo *dados.dat*.

Os registros de filmes seguem o mesmo formato usado no 1º trabalho: registros de tamanho variável, iniciados por um campo de tamanho de dois bytes (**short**), com o “|” como separador de campos. O campo IDENTIFICADOR será usado com chave primária. Portanto, o procedimento de inserção deverá cuidar para que chaves repetidas não sejam inseridas.

Para cada registro do arquivo *dados.dat*, deverá ser inserido na árvore-B o respectivo IDENTIFICADOR e byte-offset. Dessa forma, além de um vetor de chaves e um vetor de ponteiros, as páginas da árvore-B deverão conter um vetor de byte-offsets. Isso é necessário para que árvore-B seja utilizada como um índice para o arquivo *dados.dat*.

Cada equipe é livre para definir a estrutura interna das páginas da árvore, desde que a mesma obedeça às propriedades de uma árvore-B de ordem 5 e ao que é pedido nesta especificação. **A árvore-B deverá ser mantida em arquivo**, logo, a árvore nunca deverá estar **completamente** carregada na memória.

O programa deverá oferecer as seguintes funcionalidades:

- Criação da árvore-B a partir do arquivo de registros;
- Busca de registro pelo identificador do filme;
- Inserção de novo registro;
- Impressão da árvore-B.

Assim como no 1º trabalho, as operações serão realizadas sem interação direta com o usuário, utilizando para isso a linha de comando com passagem de parâmetros.

### Criação da árvore-B

A funcionalidade de criação da árvore-B (-c) deve ser acessada pela linha de comando, da seguinte forma:

```
$ programa -c
```

sendo *programa* o nome do arquivo executável do seu programa. Sempre que ativada, essa funcionalidade fará a leitura do arquivo *dados.dat* e a inserção dos IDENTIFICADORES e dos respectivos byte-offsets na árvore-B, que será armazenada no arquivo *btree.dat*. Se o arquivo *btree.dat* já existir, ele deverá ser reescrito. Note que o formato do arquivo *dados.dat* será sempre igual, porém o número de filmes do arquivo (assim como o valor informado no cabeçalho) pode variar.

Ao final da criação da árvore-B, o programa deverá apresentar uma mensagem na tela indicando se essa operação foi concluída com sucesso ou não.

Para simplificar o processamento do arquivo *dados.dat*, considere que ele sempre será fornecido corretamente (i.e., o seu programa não precisa verificar a integridade desse arquivo).

## Busca e Inserção

As funcionalidades de busca e inserção (-e) também serão acessadas via linha de comando, mas no seguinte formato:

```
$ programa -e nome_arquivo_operacoes
```

sendo *programa* o nome do arquivo executável do seu programa e *nome\_arquivo\_operacoes* o nome do arquivo que contém as operações a serem executadas. Observe que, para esse tipo de execução, os arquivos *dados.dat* e *btree.dat* devem existir. Caso algum dos arquivos não exista, o seu programa deve apresentar uma mensagem de erro e encerrar a execução.

Para simplificar o processamento do arquivo de comandos, considere que ele sempre será fornecido corretamente (i.e., o seu programa não precisa verificar a integridade desse arquivo).

## Operações

Um arquivo inicial de operações será disponibilizado no *moodle* para que você possa utilizá-lo como teste. O arquivo de operações deve possuir uma operação por linha, codificada com o identificador da operação (1 = *busca* ou 2 = *inserção*) e os respectivos argumentos. A seguir são especificados o formato e a funcionalidade de cada operação.

### Operação de busca:

1 CHAVE

- 1 é o identificador da operação de busca e **CHAVE** é a *string* com o IDENTIFICADOR a ser buscado.

Exemplo: Busca do registro de identificador 57

1 57

O IDENTIFICADOR deverá ser buscado na árvore-B usando o algoritmo de busca em árvore visto em aula. Uma vez que o mesmo foi encontrado, o byte-offset correspondente deverá ser usado para acessar o registro do filme correspondente e os dados do filme deverão ser mostrados na tela. Caso a chave não seja encontrada, apenas uma mensagem deverá ser mostrada.

### Operação de inserção:

2 REGISTRO

- 2 é o identificador da operação de inserção e **REGISTRO** é a *string* de um registro completo (em formato similar ao do arquivo de registros) a ser inserido no arquivo *dados.dat*.

Exemplo: Inserção do registro de identificador 102 e título "X-Men: First Class"

2 102|X-Men: First Class|Action|24/05/2011|132|7.1|

Note que os dados do registro serão fornecidos com os delimitadores corretos, porém o tamanho em bytes do registro a ser inserido não é fornecido como entrada e, portanto, deverá ser calculado no momento da inserção. O registro, bem como o seu tamanho, deverá ser gravado no final do arquivo *dados.dat* e o seu IDENTIFICADOR, juntamente com o seu byte-offset, deverá ser inserido na árvore-B. Como a árvore-B será usada como índice primário, chaves duplicadas não devem ser inseridas. O algoritmo de inserção em árvore-B apresentado em aula já faz essa verificação.

Considere como exemplo o arquivo de operações abaixo:

```
1 57
1 104
2 107|Shrek the Third|Fantasy|17/05/2007|93|6.0|
2 112|Transformers|Action|27/06/2007|144|6.6|
1 112
```

Esse arquivo representa a execução consecutiva das seguintes operações:

- Busca pelo registro de chave 57
- Busca pelo registro de chave 104
- Inserção do registro do filme de identificador 107 ("Shrek the Third")
- Inserção do registro do filme de identificador 112 ("Transformers")
- Busca pelo registro de chave 112

Para cada operação executada, seu programa deve apresentar na tela uma informação sobre o resultado da operação. Utilizando como exemplo as operações listadas no arquivo acima, o seu programa deve apresentar:

```
Busca pelo registro de chave "57"
57|Star Trek Beyond|Action|07/07/2016|122|6.6| (46 bytes)

Busca pelo registro de chave "104"
Erro: registro nao encontrado!

Insercao do registro de chave "107" (46 bytes)

Insercao do registro de chave "112" (43 bytes)

Busca pelo registro de chave "112"
112|Transformers|Action|27/06/2007|144|6.6| (43 bytes)
```

## Impressão da árvore-B

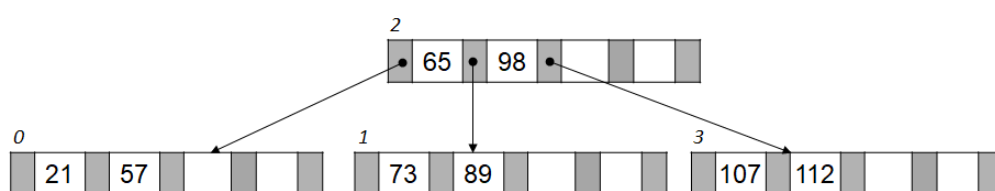
A funcionalidade de impressão da árvore-B (-p) também será acessada via linha de comando, no seguinte formato:

```
$ programa -p
```

sendo `programa` o nome do arquivo executável do seu programa. Note que, para essa execução, o arquivo `btree.dat` deve existir. Caso o arquivo não exista, o seu programa deve apresentar uma mensagem de erro e encerrar a execução.

Sempre que ativada, essa funcionalidade apresentará na tela o conteúdo de todas as páginas da árvore-B armazenada no arquivo `btree.dat`. Para cada página da árvore deverá ser informado: (a) o seu RRN; (b) os valores do vetor de chaves; (c) os valores do vetor de byte-offsets; e (d) os valores do vetor de ponteiros. Também deverá ser informado qual é a página raiz da árvore-B.

Como um exemplo, considere que a árvore-B esteja na seguinte configuração:



Note que, embora a figura mostre apenas as chaves e os ponteiros, cada página também possuirá um vetor de byte-offsets. Supondo que essa seja a árvore, seu programa deverá apresentar:

```
Pagina: 0
Chaves: 21 | 57
Offsets: 1982 | 2887
Filhos: -1 | -1 | -1
```

```
Pagina: 1
Chaves: 73 | 89
Offsets: 1101 | 1842
Filhos: -1 | -1 | -1
```

```
- - - - Pagina Raiz - - - -
```

```
Pagina: 2
Chaves: 65 | 98
Offsets: 1784 | 4
Filhos: 0 | 1 | 3
```

```
- - - - -
```

```
Pagina 3:
Chaves: 107 | 112
Offsets: 5251 | 5299
Filhos: -1 | -1 | -1
```

Quaisquer dúvidas referentes a esta especificação deverão ser sanadas com os respectivos professores.

***BOM TRABALHO!***