

1º Trabalho Prático

O 1º trabalho prático relativo à disciplina **Organização e Recuperação de Dados** constitui-se do desenvolvimento de um programa conforme as especificações abaixo.

O programa deverá ser desenvolvido na linguagem C (compilador gcc) e deverá ser feito em trios.

A entrega do trabalho será feita pelo *moodle* e deverão ser entregues todos os arquivos de código-fonte (arquivos .c e .h (se houver)).

Especificação

O arquivo *filmes.txt* (disponível no *moodle*) possui informações a respeito de filmes, armazenadas em formato *string*, sendo que os campos de um mesmo filme, separados por “;”, estão em uma única linha.

Cada filme possui os seguintes campos (nessa ordem):

1. IDENTIFICADOR do filme (que será usado como campo chave);
2. TÍTULO do filme;
3. GÊNERO do filme;
4. DATA de lançamento;
5. TEMPO de duração;
6. NOTA de avaliação do filme.

O programa deverá oferecer as seguintes funcionalidades:

- Importação dos dados;
- Busca de registro pelo identificador do filme;
- Inserção de novo registro;
- Remoção de registro.

Todas as operações serão realizadas sem interação direta com o usuário, utilizando para isso a linha de comando com passagem de parâmetros.

Importação

A funcionalidade de importação (-i) deve ser acessada pela linha de comando, da seguinte forma:

```
$ programa -i nome_arquivo_importacao
```

sendo *programa* o nome do arquivo executável do seu programa e *nome_arquivo_importacao* o nome do arquivo que contém os dados para a importação. Sempre que ativada, essa funcionalidade importará os dados do arquivo informado e criará o **arquivo de registros** *dados.dat*. Se este arquivo já existir, ele deverá ser reescrito. Note que o formato do arquivo de importação (*filmes.txt*) será sempre igual, porém o número de filmes listados no arquivo pode variar.

O programa deverá ser capaz de importar os dados do arquivo de importação e armazená-los na forma de registros de tamanho variável. Os registros devem ser iniciados por um campo de tamanho que ocupe dois bytes (*short int*) e utilizar o “|” como separador de campos.

Ao final da importação, o programa deve apresentar uma mensagem na tela indicando se essa operação foi concluída com sucesso ou não.

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

Busca, Inserção e Remoção

As funcionalidades de busca, inserção e remoção (-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 para executar as funcionalidades desejadas. Observe que, para esse tipo de execução, o arquivo de registros `dados.dat` deve existir. Caso ele 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).

Formato do Arquivo de Operações

Um arquivo inicial de operações será disponibilizado no *moodle* para que você possa utilizá-lo como exemplo. O arquivo de operações deve possuir uma operação por linha, codificada com o identificador da operação (1 = *busca*, 2 = *inserção* ou 3 = *remoção*) e respectivos argumentos. A seguir é especificado o formato de cada operação:

Operação de busca:

1 CHAVE

- 1 é o identificador da operação de busca e **CHAVE** é a *string* da chave de busca.

Exemplo: Busca do registro de chave 57

1 57

Operação de inserção:

2 REGISTRO

- 2 é o identificador da operação de inserção e **REGISTRO** é a *string* de um registro completo (no mesmo formato encontrado no arquivo de importação) a ser inserido no arquivo de registros.

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

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

Note que o tamanho em bytes do registro a ser inserido não é fornecido como entrada e, portanto, deverá ser calculado no momento da inserção.

Comando de remoção:

3 CHAVE

- 3 é o identificador da operação de remoção e **CHAVE** é a *string* da chave de busca para a remoção.

Exemplo: Remoção do registro de chave 98

3 98

Considere como exemplo o arquivo de operações abaixo:

```
1 57
1 104
2 107;Shrek the Third;Fantasy;17/05/2007;93;6.0
3 37
3 209
2 112;Transformers>Action;27/06/2007;144;6.6
```

Este 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")
- Remoção do registro de chave 37
- Remoção do registro de chave 209
- Inserção do registro do filme de identificador 112 ("Transformers")

Apresentação dos Resultados das Operações

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)
Local: fim do arquivo

Remocao do registro de chave "37"
Registro removido! (70 bytes)
Posicao: offset = 911 bytes

Remocao do registro de chave "209"
Erro: registro nao encontrado!

Insercao do registro de chave "112" (43 bytes)
Local: offset = 911 bytes (0x38f)
Tamanho do espaco reutilizado: 70 bytes
```

Gerenciamento de Espaços Disponíveis

As alterações que venham a ocorrer no arquivo de registros após a importação (i.e., inserções e remoções) deverão ser persistentes. Dessa forma, sempre que o programa for iniciado para a execução de operações, ele deverá abrir o arquivo de registros, se o mesmo existir, permitindo assim a sua edição. Durante a execução do programa, o arquivo de registros deverá ser mantido e acessado em disco.

A remoção de registros será lógica e o espaço disponível resultante da remoção deverá ser inserido na Lista de Espaços Disponíveis (LED). **A LED deverá ser mantida no próprio arquivo de registros**, conforme visto em aula, e o programa deverá implementar todos os mecanismos necessários para o seu gerenciamento e para a reutilização dos espaços disponíveis. Adote o **primeiro ajuste** (*first fit*) como estratégia de gerenciamento da LED.

No momento da inserção de novos registros no arquivo, a LED deverá ser consultada. Se existir um espaço disponível adequado para a inserção, o novo registro deverá ser inserido nesse espaço. Sobras de espaço resultantes da inserção poderão ser mantidas como parte do registro (fragmentação interna). Se na LED não houver um espaço adequado para o novo registro, o registro deverá ser inserido no final do arquivo.

BOM TRABALHO!