Trabalho prático 3 – Sugestão de Conteúdo Multimédia

1) Informação geral

O objetivo do trabalho prático 3 é avaliar a capacidade do estudante para analisar um problema algorítmico, utilizando estruturas derivadas às apresentadas na unidade curricular, e implementar em C uma solução correta e eficiente.

Este trabalho deverá ser feito de forma autónoma por cada grupo na aula prática e completado fora das aulas até à data limite estabelecida. A consulta de informação nas diversas fontes disponíveis é aceitável. No entanto, o código submetido deverá ser apenas da autoria dos elementos do grupo e quaisquer cópias detetadas serão devidamente penalizadas. A incapacidade de explicar o código submetido por parte de algum elemento do grupo implicará também numa penalização.

O prazo de submissão no Moodle de Programação 2 é 31 de maio às 23:59.

2) Descrição

As plataformas de conteúdo multimédia como por exemplo Netflix e HBO sugerem conteúdos pertinentes a cada cliente, mediante os que foram previamente consumidos. No entanto, nos servidores das plataformas referidas, existem milhões de pedidos de sugestões em simultâneo. Uma vez que o cliente não quer esperar demasiado pela apresentação das sugestões de conteúdos, cada pedido deverá ser atendido com a maior celeridade possível.

Para satisfazer estes pedidos de forma eficiente e expedita, é necessário que a informação de clientes e filmes seja guardada em duas estruturas de dados apropriadas, uma associada aos clientes e outra associada aos filmes.

A estrutura de dados que guarda os dados dos clientes está já implementada com base numa tabela de dispersão e denomina-se por `colecaoClientes` (ver "movies.h" para mais detalhes).

O registo `cliente`, que é referenciado num apontador pertencente ao registo `elementoCliente` da tabela de dispersão contém a informação relativa a um cliente:

- *Username*: combinação de caracteres alfanuméricos única para cada utilizador
- vistos: `vetor` de inteiros correspondentes aos identificadores únicos de cada filme

Por sua vez, cada `filme` terá como conteúdo:

- Título: string contendo o título do filme
- Categoria: string contendo a categoria do filme
- filmId: inteiro que corresponde a um identificador único para cada filme (sequencial que começa no 1)

- Rating: float com a classificação do filme entre 0 e 10 (sendo 10 um filme bom)

O mecanismo de sugestão de conteúdos deverá seguir uma dupla ordem decrescente. Tal como referido anteriormente, deverão ser tidos em conta os conteúdos previamente vistos pelo utilizador. Desta forma, as sugestões deverão começar pela categoria mais vista do utilizador (em caso de empate de categorias vista é o critério do que tem a média de rating mais alta) e só quando a próxima sugestão dessa categoria estiver abaixo de um determinado limiar, passar para a próxima categoria mais vista do utilizador. Por exemplo, assuma-se que o existem na coleção de filmes três filmes com rating de 9 e dois filmes com rating de 8 da categoria de comédia, quatro filmes com rating de 9 e um filme com rating de 6 da categoria drama. Assumindo que comédia é a categoria mais vista pelo utilizador, ao ser feito um pedido de três sugestões com limiar de 8.5, deverão ser apresentados os três filmes da categoria comédia com rating de 9. No entanto, se forem pedidas 4 sugestões, serão sugeridos os mesmos 3 da categoria comédia mais um da categoria drama com rating de 9.

Adicionalmente, o programa deverá suportar a inserção de novos filmes em *runtime* e novos utilizadores que uma vez mais deverão ser feitas da forma mais eficiente possível.

Para teste do programa serão pedidos vários números de sugestões diferentes para vários utilizadores, simulando um servidor respondendo a estes pedidos, os quais deverão ser atendidos da forma mais célere possível. Serão ainda realizados pedidos de adição de filmes e novos utilizadores.

Os estudantes deverão implementar:

1. `clienteAdiciona(colecaoClientes *td, const char *username, unsigned int filmId)`
2. `colecaoNumClientes(colecaoClientes *td)`
3. `clienteExiste(colecaoClientes *td, const char *username)`
4. A estrutura `colecaoFilmes` utilizando para o efeito uma estrutura de dados à sua escolha das que foram estudadas ao longo do semestre
5. `inserirNovoFilme(colecaoFilmes* colecFilmes, char* titulo, char* categoria, int filmId, float rating);`
6. `filmesCarrega(const char *nomeFicheiro)`
7. `removerFilme(colecaoFilmes* colecFilmes, colecaoClientes *td, int nFilmes);`
8. `colecaoFilmesApaga(colecaoFilmes* colecFilmes, colecaoClientes *td);`
9. `sugestoes(colecaoFilmes* colecFilmes, colecaoClientes *td, char* username, int nFilmes, float threshold)`

Para todas as funções a implementar, uma descrição da própria, dos seus argumentos e retornos poderá ser consultada no próprio *header file* "movies.h".

Os dados relativos aos filmes estão disponíveis no ficheiro **"filmes.txt"** e estão no formato: **título | categoria| filmId| rating.**

Os dados relativos aos clientes estão disponíveis no ficheiro **"clientes.csv"** e estão no formato: **username, [filmIds vistos]**. Por exemplo **hsmie483r,140,135438,3548,13498.**

3) Avaliação

A classificação do trabalho é dada pela avaliação feita à implementação submetida pelos estudantes mas também pelo desempenho dos estudantes na aula dedicada a este trabalho. A classificação final do trabalho (T3) é dada por:

T3 = 0.5 Implementação + 0.3 Eficiencia + 0.1 Memória + 0.1 Desempenho

A classificação da implementação e da performance serão essencialmente determinada por testes automáticos adicionais. Serão avaliados os outputs produzidos pelo código dos estudantes e o tempo de execução face a uma implementação de referência. No caso da implementação submetida não compilar, esta componente será de 0%. Haverá uma diferenciação da classificação, de acordo com a eficiência como explicito na fórmula.

A gestão de memória também será avaliada, sendo considerados 3 patamares: 100% nenhum *memory leak*, 50% alguns mas pouco significativos, 0% muitos *memory leaks.*

O desempenho será avaliado durante a aula e está dependente da entrega do formulário "Preparação do trabalho" que se encontra disponível no Moodle. A classificação de desempenho poderá ser diferente para cada elemento do grupo.

4) Submissão da resolução

A submissão é apenas possível através do Moodle e até à data indicada no início do documento. Deverá ser submetido um ficheiro *zip* contendo:

- ficheiro **movies.c e movies.h**
- **outros** ficheiros **.c/.h** que tenham sido utilizados como bibliotecas adicionais
- ficheiro **autores.txt**, indicando o nome e número dos elementos do grupo

Nota importante: apenas as submissões com o seguinte nome serão aceites: `T3_G<numero_do_grupo>.zip`. Por exemplo, `T3_G999.zip`

5) Alguns resultados esperados

Nota muito importante: ao compilar têm de usar **–lm** por causa da função powf da biblioteca math.h que é usada na função hash.
Exemplo: clang movie-teste.c movies.o vetor.o –lm

```
CARREGAR CLIENTES e FILMES
        Tempo carregar clientes: 0.000651
        Tempo carregar filmes: 0.000567
        colecaoNumClientes: numero de clientes correto (10)
        clienteExiste: retorno ok para cliente 'KB6OC0HI' (1)
        clienteAdiciona: retorno ok para adicionar o novo cliente 'abcde' e numero de
clientes correto (1 - 11)
        clienteAdiciona: retorno ok para adicionar um cliente que j├í exciste 'Jh' e um
filme que j├í viu '60' e numero de clientes correto (0 - 11)

SITUACAO 1 - SUGEST├öES DA MESMA CATEGORIA
        Tempo de um pedido de sugestoes: 0.000011
        Sugerido: [18 - 65 - 71 - 82 - 36]
        Esperado: [18 - 65 - 71 - 82 - 36]
        Sugestoes corretas para situacao inicial

SITUACAO 2: CLIENTE TEM CATEGORIAS EMPATADAS E FILMES T├èM RATING EMPATADOS
        Tempo de um pedido de sugestoes: 0.000009
```

```
        Sugerido: [10 - 57 - 39 - 20 - 13]
        Esperado: [10 - 57 - 39 - 20 - 13]
        Sugestoes corretas para situacao inicial

SITUACAO 3- LIMIAR OBRIGA A MUDAR DE CATEGORIA
        Tempo de um pedido de sugestoes: 0.000022
        Sugerido: [14 - 57 - 39 - 20 - 13]
        Esperado: [14 - 57 - 39 - 20 - 13]
        Sugestoes corretas para situacao inicial

REMOVER UM FILME
        Tempo de um pedido de sugestoes: 0.000025
        Sugerido: [57 - 39 - 20 - 13 - 0]
        Esperado: [57 - 39 - 20 - 13 - 0]
        Sugestoes corretas apos remocao de um filme

INSERIR UM FILME
        Tempo de um pedido de sugestoes: 0.000020
        Sugerido: [101 - 57 - 39 - 20 - 13]
        Esperado: [101 - 57 - 39 - 20 - 13]
        Sugestoes corretas apos insercao de um filme
```