



INTRODUÇÃO AOS ALGORITMOS DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

CAROLINA COSTA UDE	202420545
FERNANDO CHAVES SCARABELI	202420220
JOSÉ VÍTOR MACHADO DE OLIVEIRA	202420214

1. Introdução

O programa tem como objetivo ler um arquivo em Binário que possui 100 filmes, organizado em ranking, ano de lançamento, nome do filme, diretor e valor da bilheteria, permitindo a visualização e manipulação do mesmo. Caso não consiga ler o arquivo em binário, o programa tenta ler o arquivo CSV reserva. O algoritmo está dividido em funções com funcionalidades singulares, que, quando chamadas, fazem parte de algo complexo. Essas funções devem aparecer em um menu interativo na tela para que os usuários decidam se que querem: listar os filmes, cadastrar novos filmes, buscar ou deletar algum filme da lista, imprimir filmes por intervalo, alterar dados de um filme, salvar as alterações ou sair do programa.

2. Ordem dos dados do arquivo

Ranking (int): Posição do filme de acordo com a sua bilheteria;

Lançamento (int): ano de lançamento do filme nos cinemas;

Nome (string com espaços) : para nome do filme;

Diretor (string com espaços) : para nome do diretor;

Bilheteria (double): arrecadação total do filme;

3. Estruturas utilizadas

Redimensionamento do Vetor - “RedimensionarVetor”

O redimensionamento do vetor, realizado pela função “RedimensionarVetor”, garante que o programa possa armazenar novos registros de filmes quando o limite atual é atingido. Esse processo envolve a criação de um novo vetor com capacidade aumentada, a cópia dos dados existentes e a liberação da memória anteriormente alocada. Esse procedimento é necessário para manter a eficiência do programa, evitando erros como estouro de memória, garantindo que o sistema possa crescer conforme necessário.

Impressão de Dados - “imprime”

A função imprime é responsável por exibir as informações dos filmes de forma organizada, destacando a formatação da bilheteria, que inclui separadores de milhar para facilitar a leitura. Além disso, essa função cuida da disposição dos dados para que fiquem visualmente agradáveis e fáceis de interpretar, deixando a leitura clara para o usuário.

Busca Binária - “BuscaBinariaRecursivaPorRanking” e “BuscaBinariaRecursivaPorTitulo”

Para buscas rápidas, o código utiliza a busca binária implementada de forma recursiva nas funções citadas acima. Essa forma é eficiente, desde que o vetor esteja ordenado. A busca binária permite localizar registros específicos com maior rapidez, reduzindo o número de comparações necessárias em relação à busca linear.

Leitura de Arquivos - “lerArquivo” e “lerArquivoBinario”

A leitura de dados é feita tanto a partir do arquivo CSV quanto arquivo binário, por meio das funções lerArquivo e lerArquivoBinario. Destaca-se o tratamento de strings nos arquivos binários, que inclui a leitura do tamanho da string antes dos próprios dados, garantindo a integridade das informações. Esse processo é essencial para a correta recuperação dos dados, evitando erros de leitura.

Filtragem por Intervalo - “imprimirPorIntervalo”

A função imprimirPorIntervalo permite listar filmes dentro de um intervalo de rankings definido pelo usuário, facilitando a visualização de um subconjunto específico dos dados. Essa funcionalidade é útil para análises focadas em determinadas faixas de filmes, como os 10 filmes mais lucrativos ou então os lançados em um período específico.

Salvar Dados - “salvarArquivo” e “salvarEmBinario”

Para garantir que os dados sejam salvos, o código conta com as funções `salvarArquivo` e `salvarEmBinario`, que atualizam o arquivo CSV e binário com as alterações feitas durante a execução do programa. Essas funções asseguram que nenhuma informação seja perdida entre sessões do programa, permitindo que o usuário continue de onde parou.

Ordenação de Dados - “mergePorTitulo” e “mergeSortPorBilheteria”

Na ordenação, o código utiliza o método Merge Sort nas funções `mergeSortPorTitulo` e `mergeSortPorBilheteria`, permitindo ordenar filmes por título ou bilheteria. O Merge Sort é um algoritmo eficiente, de maior complexidade, ideal para grandes volumes de dados. Após a ordenação por bilheteria, o ranking dos filmes é atualizado automaticamente, garantindo a classificação correta da bilheteria de cada filme.

Registro de Novos Filmes - “registrarNovo”

A manipulação dos dados inclui o registro de novos filmes com a função `registrarNovo`, que insere o filme no vetor e o reordena. Esse processo assegura que os novos registros estejam sempre organizados de acordo com o critério de ordenação vigente, facilitando futuras buscas e visualizações.

Busca por título- “buscaTitulo”

Solicita que o usuário digite um título e o sistema busca o filme correspondente e reordena a lista por título antes da busca binária.

Exclusão Lógica - “DeletarFilme” e Compactação - “compactarVetor”

A exclusão lógica de filmes é realizada com a função `DeletarFilme`, que marca o registro como deletado sem removê-lo fisicamente da memória. Posteriormente, a função `compactarVetor` é responsável por remover fisicamente esses registros, otimizando o uso da memória. Esse método de exclusão é eficiente, pois evita operações de realocação constantes, concentrando a limpeza da memória em momentos específicos.

Registrar novo filme - “registrarNovo”

Permite inserir um novo filme no arquivo e manda redimensionar o vetor caso necessário.

Listar os filmes - “ListarFilmes”

Permite listar todos os filmes do arquivo, e oferece uma escolha se quer ver essa listagem de acordo com o ranking (tecle 1) ou por ordem alfabética (tecle 2).

Alterar filme - “AlterarFilme”

Permite a alteração dos dados do filme, para que seja corrigido possíveis atualizações de informações sobre o filme buscado ou algum erro que não condiz com a realidade. Basta digitar o seu ranking e o sistema permitirá a edição de cada campo específico.

Menu principal - “menu”

Essa função tem por objetivo exibir as opções disponíveis que o sistema oferece para que o usuário possa proceder sua escolha. Basta que seja teclado o número escolhido e o sistema o conduzirá. Foi usada a estrutura switch case para maior organização do código, visto que, o torna mais claro e compreensível.

4. Erros e acertos durante o desenvolvimento do trabalho

Durante o desenvolvimento do trabalho, encontramos desafios relacionados à exibição correta dos dados na função “imprime”, especialmente na formatação dos valores monetários da bilheteria. O problema estava na maneira como a parte inteira e os centavos eram tratados, o que gerava inconsistências na exibição.

Inicialmente, ao separar a parte inteira e os centavos do valor da bilheteria (double), percebemos que, em alguns casos, os centavos não eram arredondados corretamente. Isso acontecia devido a pequenas imprecisões no cálculo de ponto flutuante, resultando em valores incorretos ao converter para formato monetário.

Para resolver esse problema, implementamos um método de arredondamento mais preciso. Utilizamos a expressão “(bilheteria - parteInteira) * 100 + 0.5”, garantindo que os centavos fossem sempre arredondados corretamente para duas casas decimais. Dessa forma, valores como 0.675 passaram a ser corretamente representados como 68 centavos.

Outro desafio foi a inserção dos separadores de milhar na parte inteira do valor da bilheteria. No primeiro teste, a formatação ficava incorreta ao adicionar os pontos de separação, pois a iteração sobre a string estava desalinhada. Ajustamos o algoritmo para percorrer a string da

direita para a esquerda, inserindo os separadores a cada três dígitos, resultando em um formato correto, como "1.234.567" para 1.234.567,89.

Por fim, a exibição dos centavos também apresentou pequenos erros em casos onde o número de centavos era menor que 10, pois não aparecia o zero à esquerda (exemplo: "5" em vez de "05"). Para corrigir isso, utilizamos “stringstream”, garantindo que sempre houvesse dois dígitos na exibição dos centavos.

Após essas correções, a função “imprime” passou a exibir corretamente os valores monetários, garantindo que a bilheteria fosse apresentada de forma legível e padronizada. Esse processo evidenciou a importância da precisão nos cálculos e da atenção aos detalhes na formatação de saída, contribuindo para a melhoria da legibilidade e usabilidade do sistema.

Outro desafio encontrado no desenvolvimento foi na função “salvarEmBinario”, especialmente na consistência entre a escrita e a leitura dos arquivos binários. O problema estava na maneira como os dados eram armazenados, o que gerava erros ao tentar recuperá-los posteriormente. Inicialmente, ao gravar os valores numéricos (ranking, ano de lançamento e bilheteria), percebemos que, em alguns casos, os dados não eram lidos corretamente. Isso acontecia devido a desalinhamentos na memória e à falta de um método padronizado para interpretar os bytes gravados no arquivo.

Para resolver esse problema, implementamos o “reinterpret_cast” para garantir que os valores fossem armazenados corretamente. Utilizamos a expressão “reinterpret_cast<const char*>(&variavel)”, permitindo que os dados fossem tratados como blocos de bytes e escritos de forma consistente. Dessa forma, valores como “filme.rating”, “filme.lancamento” e “filme.bilheteria” passaram a ser armazenados sem risco de corromper a estrutura do arquivo.

Outro desafio foi a gravação das strings “nome” e “diretor”. No primeiro teste, a leitura dos nomes ficava incorreta porque o tamanho das strings não era armazenado previamente. Ajustamos o algoritmo para escrever primeiro o tamanho da string (size_t), garantindo que a função de leitura soubesse exatamente quantos bytes recuperar antes de converter os dados para string.

Por fim, a escrita das strings também apresentou problemas em casos onde o nome ou o diretor possuíam caracteres especiais ou espaços extras. Para corrigir isso, utilizamos o método “c_str()” ao escrever os caracteres no arquivo binário, garantindo que a string fosse

armazenada sem erros e sem necessidade de alocações adicionais. Após essas correções, a função “salvarEmBinario” passou a gravar os filmes corretamente, garantindo que os dados pudessem ser recuperados sem falhas. Esse processo evidenciou a importância de um formato de escrita padronizado e da atenção à conversão correta dos tipos de dados ao trabalhar com arquivos binários.

5. Considerações Finais

O algoritmo desenvolvido manipula e gerencia dados sobre as maiores bilheterias do cinema, utilizando estruturas eficientes para armazenamento, busca e ordenação. A principal estrutura utilizada é um vetor dinâmico contendo registros do tipo MBC, que armazenam informações sobre os filmes, como ranking, ano de lançamento, nome, diretor e bilheteria. Em todo o momento o algoritmo prezou pela eficiência utilizando estruturas robustas e otimizadas que se adequam ao que foi proposto, organização para maior compreensão das funções e facilidade de interação com o usuário.