

Trabalho Prático

Data de entrega: 06/06/2021 até às 23:59 pelo Canvas

Valor: 30 pontos (para alunos da CC) / 25 pontos (para alunos da EC)

Penalidade por atraso: Valor total do trabalho, ou seja, não se aceita atraso.

Trabalho em grupos de até 2 alunos

Descrição

Neste trabalho, você deverá implementar um sistema de cadastro de prontuários para uma empresa de planos de saúde, com opções de inserção, alteração, exclusão e impressão.

Especificações do programa

- O programa deve ser implementado em C, C++ ou Java e contido em um único arquivo-fonte. Todo o código deve ser de autoria do grupo, ou seja, não se aceita uso de bibliotecas de terceiros. O código deve ser comentado de modo a se compreender a lógica utilizada. A não observância deste critério implica na redução da nota final em 30%.
- A organização do arquivo de dados deve ser indexado, com índice baseado em hashing dinâmico.
- O sistema deverá oferecer uma tela inicial com um menu de opções:

1-Criar arquivo: Cria um novo sistema de arquivos constituído por um diretório, um índice e um arquivo-mestre.

- a) O diretório é um arquivo de inteiros de 4 bytes, cujos valores apontam para o número de uma página do arquivo de índices (começando por 0). O valor da profundidade global (p) deve ser armazenado no início do arquivo. Durante o processamento, o diretório deve ficar sempre carregado em memória primária mas será atualizado em disco sempre que houver alguma mudança em sua estrutura e/ou conteúdo. A profundidade global inicial deve ser parâmetro para do programa.
- b) O índice é um arquivo de inteiros de 4 bytes, organizado por *buckets*. Cada *bucket* contém o valor de profundidade local (p') seguido por n entradas, as quais são constituídas por um CPF e o número do registro no arquivo-mestre onde o registro correspondente ao CPF se encontra (começando por 0). Por simplicidade, considere que o número de CPF é um inteiro que varia de 1 a 999999999. O valor -1 indica que a entrada está desocupada. O número de entradas por bucket deve ser um parâmetro do programa.
- c) O arquivo-mestre é um arquivo de registros de tamanho fixo, contendo no mínimo os campos para nome, data de nascimento, sexo e uma área de m caracteres/bytes para anotações do médico. O valor de m é um parâmetro do programa.

2-Inserir registro: Insere um novo registro, através da entrada dos valores dos campos, exceto o campo de anotações. Os arquivos são atualizados segundo a técnica de hash dinâmico. O tempo decorrido no processo (sem envolver a interação do usuário) deve ser medido e exibido na tela.

3-Editar registro: Compreende uma busca ao registro a partir do CPF e a edição do campo de anotações. O tempo decorrido no processo de pesquisa e recuperação do registro (sem envolver a interação do usuário) deve ser medido e exibido na tela.

4-Remover registro: Remove o registro a partir do número do CPF. A remoção é sempre lógica, zerando-se o CPF no bucket correspondente. O espaço do registro do arquivo-mestre não deve ser excluído e pode ser reutilizado em uma próxima inclusão.

5-Imprimir arquivos: Imprime na tela o conteúdo atual do diretório, do arquivo de índices e do arquivo-mestre.

6-Simulação: Nesta opção deverá ser criado um conjunto aleatório de k chaves sem repetição que serão inseridos em um arquivo inicialmente vazio, com registros vazios. Em seguida, as chaves devem ser pesquisadas e os registros recuperados. Marcar o tempo de execução para a inserção e para a pesquisa do conjunto de dados. Repita o processo para diversos valores de k , p (profundidade global), n (número de entradas por bucket) e m (tamanho adicional do registro). O tamanho do arquivo-mestre deve ser maior que 1 GB em pelo menos uma das simulações. Crie gráficos para a análise do tempo de execução e analise o comportamento das funções. Realize os mesmos experimentos com os arquivos gravados em disco e em um pen-drive.
Compare os tempos observados.

Documentação

A documentação, em formato ABNT (formato Trabalho Acadêmico na página da biblioteca), deve conter:

- a) A descrição do problema, da modelagem utilizada e das técnicas implementadas para a solução.
- b) As medidas de tempo de execução para diversas exemplos de entradas e simulações, bem como a descrição do hardware e SO usados.
- c) Análise dos resultados.
- d) Referências bibliográficas.

O que entregar:

Arquivo-fonte e documentação no padrão ABNT. O nome dos componentes deve constar no arquivo-fonte e na documentação. Coloque os arquivos na raiz de um diretório cujo nome deve ser o número de matrícula de um dos componentes. Comprima o diretório e poste no CANVAS até prazo especificado. O trabalho será apresentado pelo grupo através de vídeo-conferência em data a ser divulgada.

A avaliação será baseada nos seguintes critérios:

- Correção e robustez dos programas
- Conformidade às especificações
- Clareza de codificação (comentários, endentação, escolha de nomes para identificadores)
- Parametrização
- Apresentação individual
- Documentação