

Disciplina: SCC0503 – Algoritmos e Estrutura de Dados II
Professora: Elaine Parros Machado de Sousa
Estagiário PAE: Evandro Ortigossa

Aula Prática 3 – Índices Secundários

IMPORTANTE: a entrega desta prática (completa) será utilizada, se necessário, para contabilizar presença das aulas de 03/04, 14/04 e 17/04 durante a suspensão das aulas.

Instruções:

- **Entrega individual.**
- Se desejar, utilize o código-fonte auxiliar disponível no Repositório no Tidia na pasta “Lab3”;
- A entrega deve ser feita pelo Escaninho do Tidia até o dia **20/04/2020**, às 23:55h;
- Deve ser feito *upload* em um único arquivo compactado, contendo o código fonte. O nome do arquivo compactado deve ser formado pela sigla “Lab3-” concatenado à 1ª letra do nome e concatenado ao último sobrenome do aluno. Exemplo: Lab3-esouza.zip, Lab3-eortigossa.zip;
- Escreva o nome completo e número USP no início dos arquivos de código-fonte, em comentário.
- Organize seu código utilizando comentários.

Enunciado:

Para a realização deste trabalho, você pode reaproveitar o desenvolvimento feito nas práticas anteriores para arquivos de tamanho fixo e campo fixo, e índice primário.

Você foi escalado para desenvolver uma solução que auxilie o professor a armazenar as notas de seus alunos. Para tal, implemente um programa que permita as seguintes operações em disco: inserir registros, pesquisar e apresentar registros, remover registros. Um registro conterá as seguintes informações sobre o aluno: número USP, nome, sobrenome, curso e nota. Cada aluno está vinculado a um único número USP. Todos os registros deverão ser armazenados em um arquivo de nome “**alunos.dat**”, não ordenado e sequencial (um após o outro, sem quebra de linhas). Ao final da execução do programa o arquivo deve estar atualizado conforme as operações realizadas.

O programa de manipulação de arquivos deverá ser implementado uma única versão considerando registros de tamanho fixo e campo fixo.

Boa parte dos itens solicitados aqui já foram implementados na prática Lab2.

As seguintes operações devem ser implementadas:

1. **Criação** de uma estrutura de **índice primário** (*index_prim.dat*), inicialmente vazio, junto da **criação do arquivo de dados** (*alunos.dat*). Assuma que o índice, armazenado em um arquivo em disco, cabe em RAM e portanto poderá ser manipulado em memória principal. O índice conterá a chave primária de cada registro inserido no arquivo de dados (número USP) e a posição (RRN) em que o registro foi gravado no arquivo, para **acesso direto**;

2. **Iniciar índice primário:** arquivo de índice deve ser **carregado** para uma estrutura em RAM (um vetor), ou seja, seu programa deverá ler todo o arquivo de índices para uma estrutura adequada em RAM, e somente ao final da execução reescrever o arquivo com o resultado final das operações;
3. Inserir um novo registro no final do arquivo de dados, com atualização necessária no índice primário (em RAM);
4. Criação de uma estrutura de **índice secundário** (*index_sec.dat*), inicialmente vazio. Assuma que o índice, armazenado em um arquivo em disco, cabe em RAM e portanto poderá ser manipulado em memória principal. O índice conterá como chave secundária o sobrenome do aluno. Deverá ser criado um índice **secundário fracamente ligado**, implementado com **Listas Invertidas** ;
5. **Iniciar e carregar índice secundário** a partir dos registros já inseridos no arquivo de dados e no arquivo de índice primário (carga por *bulkloading*). Lembrando que o arquivo de índice deve ser carregado para RAM para manipulação.
6. Re-implemente a função de **inserção** tal que, para cada inserção no arquivo de dados, sejam feitas as atualizações nos índices primário e secundário, se existirem. **Dica:** mantenha no cabeçalho do arquivo de dados info sobre existência ou não de arquivos de índices e respectivos nomes.
7. Pesquisar e apresentar informações de um registro usando como chave de busca o número USP; (OBS: a pesquisa deve ser feita no **índice primário**, por meio de uma **busca binária** sobre as chaves primárias. Lembre-se de manter o índice ordenado);
8. Pesquisar e apresentar informações dos possíveis registros por meio do sobrenome (ou parte do sobrenome. Lembre-se, há estratégias inteligentes para busca de palavras dentro de uma string). Exiba todos os possíveis resultados, por exemplo, para a palavra-chave “silva”, seu programa deve retornar registros como “silva”, “da silva”, etc. A busca deve ser feita via índice secundário.
9. Remover um registro do arquivo de dados usando a chave primária ou chave secundária para localização do registro. Note que a **remoção** de um registro deve ser feita de **modo lógico**, utilizando o marcador especial “*” no registro apagado (no arquivo de dados). Vale lembrar que ao remover um registro, há a necessidade de atualizar as informações correspondentes nos índices, também com **remoção lógica**;
10. Finalizar: os índices (manipulados em RAM) devem ser **reescritos** nos arquivos em disco para que, com isso, quando o programa for iniciado novamente, seja possível carregar o estado correto de edição dos arquivos; os arquivos de dados e arquivos de índice devem ser fechados ao final da execução.

Crie módulos e menu de opções para o usuário escolher as operações desejadas ou sair do programa.

Requisitos

- A estrutura de índice primário pode ser reaproveitada do exercício anterior;

- O índice secundário também pode ser manipulado em RAM, e deve ser gravado em um arquivo ao final da execução, arquivo este de onde deve fazer a leitura e carregamento para a RAM no início da execução;
- Utilizem o conceito de índice secundário fracamente ligado, que pode ser encontrado nos slides “Aula5_Arq_Indice_Sec.pdf”, disponível no repositório do Tidia;
- Seu programa deve implementar o índice secundário melhorado, ou seja, por meio de listas invertidas.
- **Não** é requerido neste exercício, remover fisicamente os registros apagados nos respectivos arquivos (dados e índices), nem desenvolver estratégias de reaproveitamento de espaço disponível ou reordenação dos índices.

Observações:

- Utilize linguagem C (ANSI);
- Caso você não tenha feito a prática anterior, não deixe de entregar esta prática! É uma ótima oportunidade para treinar seus conhecimentos em ambas as práticas, levando em conta o grande tempo disponível para entrega devido ao cancelamento das aulas presenciais;
- O código-fonte auxiliar apresenta um esboço do programa considerando registros de tamanho fixo e campo fixo. Alguns métodos básicos estão apenas sugeridos. Parâmetros e os tipos de retorno dos métodos são apenas sugestões, que podem ser alteradas se julgar necessário; outros métodos, além dos sugeridos, podem ser criados;
- Além disso, há no repositório, arquivos (.cpp) com exemplos de código para manipulação de registros que podem ser utilizados para aprendizado;
- Variáveis globais não são recomendadas;
- As bibliotecas sugeridas são o suficiente para a realização do trabalho. Não devem ser utilizadas outras bibliotecas com soluções prontas;
- Tenha em mente que índice secundário com lista invertida é uma solução sofisticada que requer tempo de pesquisa, projeto e desenvolvimento, não deixe para a última hora.
- O horário de atendimento PAE é de terça-feira, das 17h às 19h;
- Lembrem-se da política sobre plágio discutida na aula de apresentação da disciplina.