

1 Pesquisa por Cursos

Uma universidade particular possui uma tabela de valores de créditos por curso com os campos:

- código do curso (int),
- valor por crédito (double).
- nome do curso (cadeia com no máximo 300 caracteres),

Faça um programa que primeiramente digite os dados de vários cursos e os grave em um arquivo binário de nome “Cursos.dat”.

A universidade precisa acessar os cursos nesse arquivo rapidamente pelos seguintes campos: código do curso e nome do curso. Você então pensou: vou fazer pesquisa binária porque assim consigo pesquisar, no pior caso, em aproximadamente em $\log(n)$ comparações. O problema é que para fazer pesquisa binária por um dos campos, o arquivo teria que estar ordenado por aquele campo e a universidade quer rapidez de pesquisa por nome e também por código do curso. Então você pensou em fazer o seguinte: eu crio dois índices, um para códigos de curso e outro para o nomes de cursos. Cada índice é um vetor de structs. Eu crio dois tipos de struct com dois campos. Um tipo tem como campo o nome de um aluno e como segundo campo o ponteiro no arquivo onde está o aluno com aquele nome. Da mesma forma, o outro tipo é uma struct também com dois campos: o código de um curso e o ponteiro para o registro do arquivo onde o código daquele curso está armazenado. Crio dois vetores (dois índices). Um vetor do primeiro tipo e outro do segundo tipo. Ordeno o primeiro vetor em ordem alfabética de nome de aluno e ordeno o segundo vetor em ordem crescente de valor do código de curso. Quando a universidade precisar procurar um curso por código, por exemplo, aplico pesquisa binária no índice de códigos. Se a pesquisa encontrar um código no vetor de structs (código e ponteiro), eu uso o campo ponteiro correspondente, acesso o arquivo de cursos com esse ponteiro e mostro o registro do curso na tela. Faço um procedimento semelhante para fazer pesquisa por nome de curso, isto é, usando o vetor-índice de nomes.

Seu programa deve ter dois módulos(funções):

1. Módulo cadastro - esse módulo deve criar o arquivo “Cursos.dat” e ficar lendo do teclado dados dos cursos e os inserindo no arquivo até que o usuário indique que não quer mais entrar dados.
2. Módulo de criação dos índices. Esse módulo não é visto pelo usuário. O seu programa chama esse módulo logo após uma chamada ao primeiro módulo que realiza a criação do arquivo "Cursos. dat". Esse módulo deve abrir o arquivo “Cursos.dat” no modo leitura. Em seguida deve:
 - (a) Obter o valor atual do ponteiro do arquivo - existe uma função em C para isto.
 - (b) Ler o registro atual e inserir no índice de nomes o nome e o ponteiro para o registro lido e inserir no índice de códigos, o código e o ponteiro para o registro lido.
 - (c) Repetir os passos *a* e *b* acima, até que o fim de arquivo seja atingido.
 - (d) Uma vez alimentados os dois índices, o programa deve ordenar os vetores que compõem esses índices em ordem crescente.
3. Módulo de consulta - esse módulo deve perguntar ao usuário qual tipo de consulta ele quer fazer, se é por código de curso ou por nome de curso. Esse módulo fica em loop fazendo essa pergunta e só termina quando o usuário indica que não quer mais fazer consultas. Uma vez que o usuário informa qual tipo de consulta quer fazer, o módulo deve ler o valor de chave de pesquisa digitado pelo usuário e deve acionar o índice correspondente. Em seguida, deve fazer pesquisa binária no índice. Se encontrar algum elemento do índice que contém o valor da chave de pesquisa, o módulo deve usar o campo ponteiro desse elemento e acessar o arquivo em disco, usando esse ponteiro para recuperar o registro e deve mostrar o registro completo do curso na tela. Se o valor de chave não for encontrado, uma mensagem deve ser emitida indicando que não existe registro com aquele valor de chave. Terminada a pesquisa o módulo deve continuar o loop.