Relatorio TP3

Grupo 22
David Burth Kurka (070589)
Felipe Eltermann Braga (070803)
Vítor Augusto Wolf Antonioli (072622)

12 de junho de 2008

1 Descricao do trabalho

Implementamos, neste terceiro trabalho pratico, um sistema de busca por indice secundario e remocao de registros de uma base de dados, alem de manter todas as funcionalidades dos trabalhos praticos anteriores e corrigir alguns problemas (o mais serio: vetor dobravel).

Dessa forma, o programa ja conta com as seguintes funcionalidades: - insercao de registro em uma base de dados; - consulta por registro pelo nome inteiro (chave primaria); - consulta por registro por qualquer palavra (chave secundaria); - remocao de registro.

A busca por chave secundaria se da a partir da comparacao com cada palavra dos seguintes campos do registro: titulo, autor, tipo e ano. Para a remocao, implementamos um sistema de gerenciamento de memoria em disco a partir de uma lista invertida. Mais especificacoes serao dadas adiante.

2 Especificacoes do programa

Antes mesmo do usuario escolher a primeira opcao da interface, o programa faz a execucao da seguinte rotina: - aloca memoria para o vetor de registros (chave primaria + NRR); - abre (ou cria se não existir) o arquivo base.dat; - abre (ou cria se não existir) o arquivo com o indice da avail list (== -1 se nao existir avail list); - tenta abrir o arquivo pk.dat: caso exista, carrega as chaves primarias para a memoria em vetor-registros, caso contrario inicia-o a partir dos registros na base; - se a leitura de chaves primarias foi feita do arquivo base.dat, verifica se dentro do arquivo existe avail list. Em caso verdadeiro, elimina do vetor de registros os campos existentes; - cria os arquivos que representam as listas invertidas (um arquivo para cada campo / uma lista invertida para cada chave); - calcula o numero de registros trabalhados;

Depois dessa etapa, o usuario pode escolher entre as seguintes opções:

1) Insercao de registro na base de dados:

Ao ser invocada, a funcao de insercao verifica o arquivo avail_head.dat (guardamos a cabeca da lista invertida em um arquivo separado). Caso seu valor seja diferente de -1, entao ha lacunas na base, geradas pela remocao de registros. Neste caso, guarda-se temporariamente a referencia para a proxima posicao disponivel. O novo registro eh entao inserido e o arquivo avail_head.dat eh atualizado, recebendo a referencia para a proxima posicao disponivel. Caso o valor contido no arquivo seja igual a -1 (indicando fim da lista invertida), o registro eh inserido normalmente no final da base. Insere-se a chave primaria e as chaves secundarias.

2) Listagem de obras presentes no banco de dados:

Essa função mantém as mesmas especificações dos outros TPs;

3) Pesquisa por chave primaria:

Adicionada a implementacao de case insensitive e eliminacao de espacos antes e depois da chave, e entre suas palavras.

4 - 7) Pesquisa por chave secundaria:

A pesquisa por chave secundaria eh feita individualmente para cada campo do registro (titulo, tipo, autor e ano). Para cada campo, o sistema implementado baseia-se na seguinte organizacao:

Em memoria - armazena-se as chaves em um vetor de structs que contem, cada struct, a string da chave (a memoria para essa string eh alocada dinamicamente, no tamanho da palavra, sem a necessidade de alocar mais espaco do que o necessario) e o endereco (inteiro) de uma posicao no arquivo da lista invertida. Observacao importante: esta lista de chaves secundarias e enderecos tambem eh armazenada em um arquivo (sks.dat), a fim de que nao se precise gerar o vetor a partir da base toda vez). Em disco (arquivos das listas invertidas) - cada arquivo do tipo li_¡campo¿.dat contem listas invertidas, uma lista para cada chave secundaria. Esta lista eh composta pela(s) chave(s) primaria(s) (portanto, de tamanhos fixos) que correspondem a chave secundaria procurada.

Desse modo, quando o usuario procura por um titulo, eh feita uma busca binaria em memoria, que pode encontrar a chave no vetor. Sendo este o caso, havera um endereco para o arquivo da lista invertida, sendo possivel assim ter acesso a todas as chaves primarias que correspondem aos registros contidos na busca daquela chave secundaria. No caso da remocao, optamos por nao remover as estruturas de chaves secundarias. Desse modo, quando se busca uma chave secundaria, eh possivel que ela exista na estrutura, mas quando eh feita a busca pela chave primaria correspondente ela nao eh encontrada (pois a chave primaria foi removida da lista invertida), dando o retorno para o usuario de que a palavra procurada nao existe na base.

8) Remocao de registro:

A chave primaria do registro que se deseja remover eh lida e procurada no vetor que contem as chaves primarias da base. Caso seja encontrada, a funcao retorna o NRR correspondente ao registro. Atualiza-se entao a avail list da base (esta lista funciona como uma pilha). O valor da cabeca da avail eh inserido no inicio do registro na base e a cabeca eh atualizada (o NRR do

registro removido eh inserido no arquivo avail_head.dat). A chave primaria eh removida. As chaves secundarias referentes aquele registro NAO sao removidas, apenas a chave primaria (referente ao registro removido) eh removida da lista invertida daquela chave. Desse modo, mesmo que o programa encontre a SK, nao encontrara a PK do registro removido ao percorrer a lista.