

Contents

Seção 1 - Leitura de dados	2
csv2bin	2
Implementação	2
Seção 2 - Recuperação de todos os registros	2
bin2out	2
Seção 3 - Busca por critério	3
bin2outGrep	3
Decisões de projeto	3
Seção 4 - Busca por RRN	3
bin2outRRN	3
Seção 5 - Remoção lógica de registros	3
Seção 6 - Inserção de novos registros	3
Seção 7 - Atualização dos campos do registro	3
Seção 8 - Compactação (desfragmentação)	4
Seção 9 - Pilha dos RRNs removidos	4

Seção 1 - Leitura de dados

csv2bin

A função lê uma linha de cada vez do arquivo csv com freadline e separa cada campo com split e escreve os dados em outro arquivo binário (ambas funções implementadas em utils.c). Uma mensagem de erro é printada caso o arquivo de entrada e/ou saída não consiga ser aberto.

Começamos escrevendo o registro de cabeçalho no arquivo de dados. Para cada linha lida, separamos os campos de cada registro usando a função split. Os dados então são escritos no arquivo de dados. Primeiramente são escritos os campos de tamanho fixo. Para os campos de tamanho variável, usamos indicador de tamanho. Após escrever todos os dados, os arquivos são fechados e uma mensagem de sucesso é exibida na tela.

Implementação

Como estamos usando registros de tamanho fixo, decidimos que o último campo do registro (no caso 'prestadora' - ocupará o que sobra do tamanho do registro, ie, se o registro tem 87 bytes e os demais campos ocupam 50 bytes, o campo prestadora ocupará $87 - 50 = 37$ bytes. Apenas os primeiros bytes do campo realmente serão utilizados (de 2 a 4, no máximo), os demais serão espaço em branco.

Seção 2 - Recuperação de todos os registros

bin2out

A função recupera os dados de todos os registros - os que não foram removidos - do arquivo de dados 'output.dat' e os imprime de maneira organizada na tela. O arquivo de dados é aberto para leitura. Caso o arquivo não for encontrado, uma mensagem de erro é apresentada. Como não iremos alterar o arquivo de dados, apenas ignoramos o registro de cabeçalho - pulamos 5 bytes do começo do arquivo. Os dados dos registros começam a ser lidos. Se o campo 'codINEP' de um registro for -1, esse registro foi removido. Pulamos para o registro seguinte. Caso o registro não tenha sido removido, ie, campo 'codINEP' diferente de -1, recuperamos os demais campos do registro. Utilizando a função catReg (utils.c), os dados recuperados são exibidos na tela. Toda memória alocada é liberada e o arquivo de dados é fechado.

Seção 3 - Busca por critério

bin2outGrep

Começamos abrindo o arquivo de dados para leitura e assim como na função bin2out iremos recuperando todos os campos dos registros armazenados.

Essa função recebe como parâmetro o campo e o valor a ser buscado. Para cada registro recuperado, comparamos o valor do campo buscado com o do registro. Caso for igual, imprimimos os dados do registro.

Repete-se até chegar no fim do arquivo de dados.

Caso a busca não encontre nenhum registro ou caso o registro tenha sido removido, uma mensagem de erro é exibida na tela.

Decisões de projeto

Para facilitar, usamos uma função de comparação que consegue comparar tanto números quanto strings. Assim que chamamos a função bin2outGrep, já passamos como um dos parâmetros a função de comparação correta.

Seção 4 - Busca por RRN

bin2outRRN

Seção 5 - Remoção lógica de registros

marcar registro como inutilizado - * - e armazenar rrn na pilha de registros removidos

Seção 6 - Inserção de novos registros

inserir primeiramente nos espaços vazios

Seção 7 - Atualização dos campos do registro

procurar pelo registro: se existir, substituir, caso contrario mostrar mensagem de erro

Seção 8 - Compactação (desfragmentação)

Começamos renomeando o arquivo de dados fragmentado para `output.dat.old`. Essa operação poderá falhar caso não exista um arquivo de dados anteriormente. Neste caso, imprimimos uma mensagem de erro.

Criamos um novo arquivo de dados, chamado `output.dat` e inicializamos o registro de cabeçalho.

Lê-se um registro do arquivo de dados fragmentado (`output.dat.old`): Caso tenha sido removido, ou seja, caso o `codINEP` do registro seja -1, ignorar e pular para o próximo registro. Caso não tenha sido removido, recuperar os demais campos do registro e escrevê-los novamente no novo arquivo (`output.dat`). Após recuperar todos os registros, atualizar o registro de cabeçalho, indicando que os dados estão consistentes.

Imprimir uma mensagem na tela informando que a compactação foi bem sucedida.

Seção 9 - Pilha dos RRNs removidos

Função que imprime a pilha de registros removidos.