

CAPA

USP – Universidade de São Paulo

Professora Dra. Cristina D. A. Ciferri

Trabalho de SCC0215 – Organização de Arquivos (Parte 1)

Pedro Pastorello Fernandes – NUSP 10262502

Data de finalização do trabalho: 17/05/2018

CASO 1: Leitura de registros:

Na leitura de registros, é inicializado um vetor de registros que são lidos a partir do arquivo .csv pela função `leArquivoCsv()`. Depois, eles são escritos em um novo arquivo binário, assim como especificado.

CASO 2: Recuperação de dados:

Le os registros do arquivo binário e imprime eles na tela de acordo com a formatação pedida. É usada a mesma função de busca, no modo de operação "0" (retorna todos os registros).

CASO 3-4: Busca de registros:

Busca todos os registros que satisfazem a busca, utilizando a função `buscaRegistro()`:

- 0 = nenhum (printa todos os registros);
- 1 = `codEscola`;
- 2 = `dataInicio`;
- 3 = `dataFinal`;
- 4 = `nomeEscola`;
- 5 = `municipio`;
- 6 = `endereço`;
- 7 = `RRN`.

Em casos de campos do tipo string de tamanho variável, não diferencia maiúsculos e minúsculos e considera a busca de substrings em strings.

CASO 5: Remoção lógica de registros:

Remove os registros usando a função `removeRegistro()`, que utiliza uma pilha para organizar os RRNs dos registros removidos para futuras inserções.

CASO 6: Inserção dinâmica de registros:

Le os dados inseridos por argumento na chamada do programa e os organiza em um registro. Depois, os insere no arquivo de dados, na posição do topo da pilha.

CASO 7: Atualização de registros por RRN:

Le os dados inseridos por argumento na chamada do programa e os organiza em um registro. Depois, atualiza o registro do RRN especificado na chamada do programa.

CASO 8: Desfragmentação (compactação) do arquivo de dados:

Le todos os registros válidos e os armazena em memória principal, depois os escreve em um novo arquivo, que substitui o arquivo original.

CASO 9: Printar pilha de RRN vazios:

Emprime a pilha de RRNs vazios, lendo os valores do arquivos a partir da topo, que está armazenado no cabeçalho do arquivo como especificado.

Compilação: a compilação do programa deve ser feita pelo comando *make all* e, para execução, a sintaxe *./trabalho1 argumento1 argumento2 argumento3 ...*. A sintaxe dos argumentos é exatamente a especificada na especificação do projeto.