

Universidade Federal da Bahia
Graduação em Ciência da Computação
MATA54 - Estruturas de Dados e Algoritmos II
Terceiro Trabalho Prático
Prof. Flávio Assis
Semestre 2018.2 - 28 de novembro de 2018

Ordenação

1 Descrição Geral do Trabalho

Neste trabalho cada equipe de alunos deverá implementar um algoritmo para ordenação de registros. O programa deverá ler uma sequência de registros de um arquivo e gerar um novo arquivo com os registros ordenados segundo uma chave.

O arquivo de entrada conterá uma sequência de dados sobre pessoas (*registro*). Cada registro conterá: um código (chave, de valor inteiro), uma idade (valor inteiro) e nome (sequência de até 20 caracteres, contendo apenas letras minúsculas, sem acento e sem cedilha). O arquivo de entrada terá formato de texto. Cada linha conterá os dados de um registro: inicialmente o código, seguido de um espaço, seguido da idade, seguida por um espaço, seguido pelo nome.

O arquivo de saída deve também ter formato texto, com os registros ordenados, um em cada linha. Ou seja, cada linha deve conter o código de um registro, seguido de um espaço, seguido da idade, seguida de um espaço, seguido do nome. O nome deve conter uma letra como último caractere (ou seja, não gere na saída espaços após o nome). Não pode ser gerado nenhum caractere a mais na saída.

Observações:

1. O arquivo com os registros de entrada será chamado *entrada.dat*. O arquivo de saída deve ser chamado *saida.dat*.
2. O programa deve conter duas constantes: *MAXNARQS*, que indica o número máximo de arquivos que o programa pode manter abertos ao mesmo tempo; e *MAXMEM*, que indica o número máximo de *bytes* que o programa usará da memória principal. O programa deve ser escrito considerando-se esses limites. Estas constantes devem ter valor inicial 100 e 512MB (512 mega *bytes*).
3. Trabalho em equipe de até 4 alunos.

4. Deve-se assumir que o arquivo a ser ordenado pode não caber na memória principal. Não serão aceitos trabalhos que dependam de que todos os registros caibam na memória principal.
5. Podem ser usadas bibliotecas para: executar funções básicas em arquivos (abrir, fechar, ler entrada, gerar saída, modificar a posição corrente de leitura/escrita no arquivo) e manipulação de estruturas básicas em memória principal. Não pode ser usado código não implementado pela equipe cujo fim seja processar dados em memória externa.
6. O trabalho pode ser escrito em uma das seguintes linguagens: C (compatível com gcc), C++ (compatível com g++), Java ou Python.
7. A equipe deve entregar um relatório sobre a forma utilizada para ordenar. Em particular, este relatório deve conter a forma específica de uso da memória principal para ordenar e maneiras de parametrizar o uso desta memória para melhorar a eficiência do programa. O relatório deve indicar também qual foi o papel de cada membro da equipe. Todos deverão ter trabalhado na geração de código.
8. Será feito um teste dos programas com um arquivo de entrada especial. A equipe cujo programa ordenar esse arquivo mais rapidamente obterá um ponto extra. Somente participarão deste teste os programas que seguirem estritamente o formato de saída indicado acima.
9. **Data de entrega:** 16/12/2018 (prazo firme)