



Universidade de São Paulo - USP
Instituto de Ciências Matemáticas e de Computação – ICMC
Departamento de Sistemas de Computação – SSC
SSC0112 Organização de Computadores Digitais I
1º Trabalho – disponibilizado em 09/04/2013

Implementação de um quicksort recursivo em assembly ICMC

O objetivo deste trabalho é implementar em *assembly* ICMC o algoritmo Quicksort recursivo, o qual ordenará em ordem crescente um vetor de números inteiros positivos. O vetor de inteiros a ser ordenado estará fixo no código fonte e deverá ser alocado na memória do computador quando o processo for carregado no Simulador. O tamanho do vetor deverá ser definido por um rótulo, o qual deverá ser utilizado para representar o tamanho do vetor durante a definição do mesmo e demais instruções que necessitarem desta informação. O vetor original (não ordenado) deverá ser mantido na memória até a finalização do processo. Como saída, a função *main* do processo deve imprimir no vídeo o vetor não ordenado e o vetor ordenado em ordem crescente. Os vetores a serem impressos devem ser identificados no vídeo como "não-ordenado" e "ordenado", respectivamente. Cada elemento desses vetores deve ser separado por uma vírgula. O algoritmo a ser implementado deve considerar vetores de qualquer tamanho entre 1 e 20 elementos. Vetores com mais de 20 posições devem ser detectados e a execução deve ser interrompida. Cada execução do processo ordena uma única vez o vetor desordenado e então finaliza.

O código fonte deve conter uma função *main*, a qual chama uma função *quicksort*, passando como argumentos de entrada o endereço da primeira posição do vetor, a posição inicial do vetor a ser considerada e a posição final do vetor. Quando a execução for correta, a função *quicksort* retornará o endereço de um novo vetor, este contendo os elementos ordenados. Quando a execução da função *quicksort* detectar algum erro (quantidade errada de elementos a ordenar, por exemplo), retornará 0.

A função *quicksort* não poderá imprimir nada na tela. A função *main* será a responsável por imprimir os resultados finais do processo, conforme já detalhado.

O pivô a ser escolhido a cada iteração deve ser sempre o primeiro elemento do vetor (ou subvetor) a ser ordenado. O algoritmo do *quicksort* recursivo não está detalhado nesta especificação. Este algoritmo deve ser pesquisado na literatura disponível que aborda estrutura de dados e métodos de ordenação.

A correção levará em conta a execução correta do algoritmo e a qualidade do código fonte feito (identação, comentários corretos, nome significativos para rótulos, espaços entre porções do código, entre outros). Para verificar a execução correta do algoritmo é necessário que a entrada e a saída sigam, ambas, um padrão rígido:

- (a) para a entrada será considerado um vetor de inteiros fixo na memória do computador;
- (b) a saída deve ser impressa no vídeo pela função *main*, seguindo o já determinado.
- (c) o programa encerra após a execução da operação solicitada.

O código desenvolvido deverá executar tanto no Linux quanto no Windows.

Use como casos de teste, vetores de tamanho variado entre 1 e 20 elementos, vetores totalmente ordenados e vetores totalmente/parcialmente desordenados. Use também vetores de tamanho superior a 20 elementos. Nos casos de entradas inválidas, o seu algoritmo deve fornecer uma saída válida, indicando o erro e então finalizando.

A data máxima de entrega deste trabalho será fixada em sala de aula. A entrega será via Moodle da USP.

Este trabalho deverá ser feito em grupo, o qual já foi determinado no início do semestre letivo. Envie apenas um arquivo por grupo contendo o código fonte. Este arquivo **deve** ter o nome: **TY-GXX-*nnnn*** (onde **Y** indica a Turma (1 ou 2), **XX** indica o nr do grupo e ***nnnn*** indica o nome de um dos integrantes do grupo). Forneça, obrigatoriamente, como comentário no corpo do arquivo submetido o número da turma, o número do grupo e o nome de **todos** os integrantes do grupo que efetivamente participaram do desenvolvimento.

Você deve seguir rigorosamente toda a especificação deste texto a fim de validar a sua nota. Essa validação será feita por um fator de multiplicação à nota final do Trabalho: 0 (zero) por não seguir a especificação ou 1 (um) por seguir.

Quaisquer dúvidas/erros/características omissas a esta especificação deverão ser reportadas ao professor e alunos PAE para orientação de como proceder na construção e entrega do algoritmo.