

COMPARAÇÃO DE TEMPO DE EXECUÇÃO DE ALGORITMOS EM COMPILADORES DIFERENTES QUE RETORNAM O MAIOR E O MENOR ELEMENTO DE UM VETOR

Luan Roger Santos Santana¹, luanengcomputacao@hotmail.com; Ricardo de Oliveira Muniz Junior², Marcio O. Costa³, costafilo@hotmail.com.; Walteno M. Parreira Junior³, waltenomartins@yahoo.com

^{1,2} Graduandos de Engenharia da Computação. UEMG/FEIT

³ Orientadores. Curso de Engenharia da Computação. UEMG/FEIT/ISEPI

A pesquisa descreve sobre os três tipos de algoritmos utilizados para encontrar o maior e o menor elemento de uma lista de um vetor com quarenta posições, segundo Parreira Júnior, um algoritmo maxmin1 trivial pode ser melhorado apenas mudando uma comparação, obtendo o algoritmo maxmin2. E reformulando o algoritmo obtém se um novo algoritmo Maxmin3. Foram utilizados três compiladores diferentes para ver se encontram resultados significativos de velocidade na compilação, o DEV C++, para o Windows, o Borland, que é shareware, e o Anjuta para Linux. O problema consiste em que, os algoritmos executados em compiladores diferentes para ver se encontram resultados significativos de diferença de velocidade na compilação em compiladores diferentes, ou se apenas o código é suficiente para deixar um algoritmo mais veloz. Foi usado um recurso para atrasar a execução, denominado “delay”. O primeiro compilador utilizado, o DEV C++, mostrou um tempo muito bom na execução dos três algoritmos, em especial o algoritmo Maxmin3, que apresenta uma menor taxa de comparações, no pior, médio e melhor caso, executou em 0.22 segundos um total de 42 comparações, já no segundo compilador o Borland, mostrou um resultado maior, com 42 comparações ele obteve um tempo de 1,21 segundos, cinco vezes mais que o DEV C++, já o Anjuta mostra o mesmo tempo que o DEV C++, 0.22 segundos para as 42 comparações. Alguns compiladores, que oferecem mais recursos acabam se tornando inviável no quesito de tempo já que necessitam de mais processamento para serem tão eficientes quantos outros que podem ser mais leves, porém, eles acabam oferecendo os recursos básicos de forma mais intuitiva, mostrando os erros encontrados de forma, onde com alguns cliques podem ser corrigidos, e nos mais leves acabam sendo menos intuitivo, para aprendizagem, o melhor, sempre será o que oferece mais recursos, tornando inviável o rápido que não oferece recursos. Compiladores podem sim, alterar a velocidade da execução dos algoritmos, apenas melhorando o código, pode não ser suficiente para otimizar um algoritmo, quando compiladores mais leves entram em ação eles conseguem executar algoritmos com mais velocidade, dependendo do algoritmo e da finalidade, isso pode até ajudar, como por exemplo, algoritmos feitos educacionalmente, já que talvez precisa-se de algoritmos grandes, que demorem a ser executado, com muitas contas, se o compilador for lento, ele pode levar até cinco vezes mais tempo, que um compilador eficaz. Levando em consideração também, existem compiladores que por mais que sejam leves e rápidos, não possuem funções que um compilador mais lento tenha que agiliza a busca de funções especiais para os códigos, assim como as bibliotecas, que vêm inclusas, já nos compiladores mais leves, devem ser adicionada-as manualmente.

Área: Ciências Exatas e dos Materiais.

Para referenciar:

SANTANA, L. R. S., MUNIZ JUNIOR, R. O., COSTA, M. O., PARREIRA JÚNIOR, W. M. Comparação de Tempo de Execução de Algoritmos em Compiladores Diferentes que Retornam oMaior e o Menor Elemento de um Vetor In: II Mostra e II seminário de Ensino, Pesquisa e Extensão da FEIT, 2010, Ituiutaba. **Anais**. Ituiutaba: FEIT, 2010.