CSI-370-01 Computer Architecture[1] [2] Research Project 1

Tony Calarese
Champlain College
anthony.calarese@mymail.champlain.edu

For this research assignment I will be working on the speeds of sorting algorithms in **32 bit MASM Assembly**, **64 bit MASM Assembly**, and **Visual C++ code**. For this project I will incoorporate C++ and MASM assembly to work together in order to determine how the run time of a specific sorting algorithm varies with each implementation.

The purpose I want to explore is which methodology serves best with on hand experimentation. This will be a software implementation.I intend to attempt coding in C++ first and match the speeeds with that of **Quick Sorting**, **Insertion Sorting**, and **Selection Sorting**.

Some of the challenges of this project will be memory handling, implementing all architectures in one executable, and organization will be key given that this code will be lengthy to an extent. It will Also be Challenging to make sure that the sorting algorithms will work in assembly first hand; but, regarless I believe I can find a solution through studies of psuedo code and my own code.

I intend to solve the memory problems given that a dynamic array must be created for the randomly defined array. Given that one cannot hard code a true random array. However, I could hard code an array; but, given that the results are to be analyzed in mass quantaties, it would be too time constraining and too long to implement in this methodology. So , by using a dynamic pointer array, I must remember to delete the memory to prevent memory leaks.

With the means for making sure that data is without fault, everything must be taken into consideration, such as the system specs and the memory with the processor. Also making sure to analyze the data and make sure that they are being handled in similar prospectives as they would be in all context.

With the different Architectures however, that will complicate how things are compiled and ran. More than likely to avoid this I will need to source this out to two files/projects, one for 32 bit and one for 64. That is unless I find a means for letting them work together. But intil then it will more than likely need to be seperate for the time being. I think that this error will be more of a nucience than it will be a problem, due to the fact that I am more-or-less copying and pasting the C++ end which will control the Timer System and a GUI for easy user interaction.

Implementation methods

- Algorithm Implementation Median
  - Visual Studio 2017 Community
- Code Architecture/Impelentation Methods
  - Visual C++
  - 32 bit MASM assembly
    - Inline Assembly
    - Externally handled
  - 64 bit MASM assembly
    - Externally handled
- Quick Sorting [3]
  - Best Case
    - O = (n log(n))
  - Worst Case
    - O =(n log(n))
  - Average
    - O = (n^2)
- Selection Sorting [3]
  - Best Case
    - O = (n)
  - Worst Case
    - O = (n^2)
  - Average
    - O = (n^2)
- Insertion Sorting [3]
  - Best Case
    - O = (n)
  - Worst Case
    - O = (n^2)
  - Average
    - O = (n^2)

In all I am hoping to accomplish why low level computing may be better in speeds as oppposed to higher level languages, in context to what architecture is best for speed. I am going into this assignment with the full expectation that the C++ code is going to be the slowest out of the methodologies. With that said the 64-Bit Assembly seems like it would be more favorable to win in this circumstance, but this scenario is more on the lines of 'does the circcumstances matter for low level computing speeds?'

I expect to run into curveballs during this experimentation; and, I think they are going to be because of memory problems and given that I cannot multitask on my computer to make sure that all the results are efficient, it will be time consuming and difficult to process. But I believe it is doable with some patience and close observations.

References

[1] Brian R. Hall *Research1.pdf*
https://champlain.instructure.com/courses/903034/assignments/9499138

[2] Bibliography Source of Reference
https://www.overleaf.com/learn/latex/Bibliography_management_with_bibtex

[3] Big O Cheat Sheet Array Sorting Algorithms
http://bigocheatsheet.com/

[4] Math Formulas in Latex
https://www.authorea.com/users/77723/articles/110898howto-writemathematicalequationsexpressionsandsymbolswithlatexa-cheatsheet