CITS3402 and CITS5507 High Performance Computing
Laboratory Sheet 1

How to compile and run C programs in your computer

- You can write C programs using a simple editor if you have a linux or MacOS system. You can also use IDEs (Integrated Development Environments), however you have to resolve any issues with IDEs on your own. This is a third year and masters level unit and I would expect a high level of independence from you.

- If you have a Windows system, there are ways of using linux from the Windows OS, for example here and there are many other suggestions on the Internet that you can explore.

- The simple way to compile a C program from command line is to use the gcc compiler from the command line. For example, if your program file is myProg.c, you can use gcc myProg.c to generate an executable file called a.out and execute it. You can also give a specific name to your executable file, for example gcc -o myProg myProg.c will create an executable called myProg.

- Don't despair if you don't know the C language. I will cover it in the lectures quickly, and would expect you to pick up based on your experiences with other programming languages. You can also look up the lecture notes for Lecture 2 for CITS2002 that I am using as a refresher for C.

- Note: The use of gnu c compiler is identical in the Pawsey supercomputer. You can run small C programs at your login node there. However, we have to use the scheduler for submitting large jobs. I will explain soon.

Timing of C programs

- When we try to improve the run time performance of our programs, we have to know how much time our program takes to execute. C has several ways of determing the execution time of programs. Your first

task is to read this discussion and understand how you can determine the execution time of your C programs.

- While stackoverflow is a very useful discussion site for C programming, it is your responsibility to check whether the discussed codes are correct or not. You can also use other sources from the internet for your understanding.

- Your computer is so fast that you need to do a lot of computations for the time to be measurable. First try to write simple loops that execute many times,e.g., start with several hundred thousand iterations and increase the number of iterations gradually. You need not have any statements inside the loop.

- Now plot the running times against the number of iterations in a graph. You can use Microsoft Excel. Does the time increase linearly? In other words, is the increase in time proportional to the number of iterations?

- Now write a program that does something non-trivial inside a loop. For example, declare a large enough array and store random floating point numbers between 0 and 1 in the array. Now add these floating point numbers in the loop. You can see this discussion for random number generation.

- Write the bubble sort algorithm in C and check timing with larger and larger input sizes. There are many sources that you can look up for the bubble sort algorithm, starting from the Wikipedia article.

- Write the quicksort algorithm in C and check timing with larger and larger input sizes. There are again many sources that you can look up for the quick sort algorithm, starting from here. However, you have to modify this code to input large arrays.

- There are many other sorting algorithms like merge sort and insertion sort. Search the internet for C codes for these algorithms and do a timing analysis with varying input sizes. Which algorithm is the fastest?

- Note: There is a limit of how large an array you can allocate in a C program statically. This is because there is a limit on the size of an activation record on the stack (I will explain the runtime stack in the

third lecture). Most of the time you will get a run-time "segmentation fault" error if you allocate too large an array. You have to experiment and see how large an array you can allocate without segmentation fault. This situation can be avoided if one allocates arrays dynamically, since that storage is allocated in the heap.

Amitava Datta
August 2024