CS 251: [Code Warrior:] Lab 09 gdb, time, and pyplot (Inlab)

- Handed out: 10/05 2:00 PM - Due: 10/05 4:40 PM
- Please write (only if true) the honor code. If you used any source (person or thing) explicitly state it. You can find the honor code on the web page.

Objective

gdb is the standard debugger for the GNU operating system. It facilitates tracing and altering the execution of computer programs. The user can monitor and modify the values of programs internal variables, and even call functions independently of the program's normal behavior. You may also wonder why one may want to use gdb in the world of IDEs (next lab, Lab 10). One answer is that you want to use a stand alone editor (such as emacs) rather than the one the IDE provides. Second, you want to inspect the memory around the point you have some issues with in a local manner.

matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy.

Background

- Visit the url http://www.tutorialspoint.com/gnu_debugger/ and read about gdb from here to have a fair idea about it. Once you got that, see https://blogs.oracle.com/ksplice/entry/8_gdb_tricks_you_should
- Visit the url http://www.tutorialspoint.com/c_standard_library/c_function_ctime.htm to get introduction of ctime.
- Visit the url http://matplotlib.org/users/pyplot_tutorial.html to get an idea of matplotlib and pyplot.

Note

Problem 1 is harder than the second and third. Although the changes in Prob 1 are minimal, you can get frustrated debugging.

Tasks

1. [gdb] On many occasions, we are required to work on some one else's code base. The code may contain errors that needs to be traced down. We will be doing an exercise using gdb.

   You are given an erroneous code that attempts to create a linked list containing numbers 1,2,3 ..., and further reverse the list in $O(n)$ time. (By reverse we mean that traversing the link from the head results in ..., 3, 2, 1, i.e., the numbers in descending order.) At the end of the execution of the program, we should get the numbers in descending order on the standard output.

You can probably write a better “reverse" code, but your task in this inlab is different. It’s to correct the code (which has not been documented – shudder – have you ever seen that happen?!!).

The erroneous code `err.c`. Use `gdb`, or the gui front end, `ddd` to spot the errors. (You can also use `emacs` as a front end for `gdb` instead of `ddd`). There are two logical errors present in the code. The location of the first error is provided within the code itself. After you fix the errors, make sure to document the addresses of the first three nodes in main memory. You will need the gdb commands: `print, break, next`. Both programs are to be compiled with `gcc`, not `g++`.

Hints: Set break points in the `for` loop to spot the first error. The second error is a bit more involved. You may want to reduce the number 7 in the `for` loop. In gdb, to print: if `p` is a pointer, you can type `p->val` or `p->next` i.e., use variables just like in the program.

`[Note]` You are neither allowed to change any function definition, nor the logic of the program. Further keep in mind that although you can find problems such as the one in `err.c` by inspection (in this case), it’s good to learn the basic gdb commands on small pieces of code before jumping into large code.

2. `[time]` `fibo.cpp` is a simple program which calculates the n^{th} term of Fibonacci series using recursion. n is passed as command line argument. (You may need the `--std=c++0x` to compile this).

    - Learn how to use time command – both the built in shell keyword, and the program `/usr/bin/time`. Use it with `fibo` and explain the difference between the times you get.
    - Learn how to time a program inside your code using the `ctime` header. Time `fibo`. Update the file `fip.cpp` file so that it displays the output as <nth term>,<result>,<time taken> (in seconds).

3. `[pyplot]` Create a bash script `create_data.sh` which runs (the updated) `fibo.cpp` for n from 1 to 46 and generate a csv file `data.csv`. Write a python script `plot.py` to generate a plot of n and time taken, using pyplot (matplotlib).The graph should include the proper `x` and `y` labels as well as an appropriate legend.

Submission Guidelines:

Submit the following documents (and don’t submit things that can be generated).

1. For Task 1, submit the `gdb` log in a text file named `log.txt` and submit the corrected code as `correct.cpp`. Do not forget to provide the addresses of the nodes mentioned in readme file.

2. For Task 2, part 1 explain the observations in the readme file.

    Also submit updated fibo.cpp.

3. For Task 3, submit `create_data.sh` and `plot.py`

Do not forget to put `readme.txt` file in a folder. The folder and its compressed version should both be named `lab09_groupXY_final`. Hence, you submit a tar.gz named `lab09_group03_inlab.tar.gz` if your group number is 3.

How We Will Grade You

- For task 1, partial marks will be there for tracking both the errors (in the form of the corrected code) as well as for the log.
- For task 2, marks will be based on the correct observations in part (a) and for the correctness of the code for part (b).
- For task 3, marks will be based on scripts create_data.sh and plot.py as well as correctness of the plot.