PHYSICS 2T C Programming Under Linux

Linux Lab 6 GDB

INTRODUCTION

This is Lab session is about familiarising yourself with GDB, the GNU debugger and learning how to use it in developing C code. All the information you need to work with GDB can be found in the lectures which should be used as a reference.

As with previous labs, if you get stuck don't hesitate to seek help from lab demonstrators, lectures and other students. As before please supply your answers in and answer.txt file.

GETTING STARTED

Start by opening the Terminal application. Verify that you are in your home directory, you can do this by typing **pwd** (Print Working Directory) into the terminal. It should be similar to **/home/0800890**.

To obtain all the files required for this lab, please enter the following into your BASH shell:

git clone https://bitbucket.org/glaphysp2t/linux-lab06.git

Confirm you have the necessary files by entering the command: **ls linux-lab06**

QUESTIONS

1. In the **linux-lab06** directory is a modified version of the code you have seen in previous Lectures along with a makefile.
   a) Examine the Makefile, what is missing from it that is required to begin debugging our program? **Hint:** something needs to be passed to the compiler to generate debug information. Modify the Makefile and run **make debug**.
   b) Try running the program from the command line, what happens? How would we run the debugger? Do so now.
   c) How would you set a breakpoint at the beginning of the code? Set that breakpoint and begin running the code with the run command.
   d) Which line of code and which function is now highlighted/printed after execution has halted?
   e) How would you advance the execution of the code by one instruction so that you entered into the function you identified above? Do that now.
   f) Continue to run the code using **c** or **continue**. The program will now break, what is the error message and on which line did it occur?

g) How would you print out the contents of the **dataptr** variable, and what is its contents?

h) How would you print out the contents of the memory location it points to, what happens when you do so? From reading the error message what do you think the problem with this section of the code is?

i) Quit out of gdb using the **quit** command. Change line 6 in the **util.c** file to read:

int *dataptr = data;

Re-compile the code and try running from the command line. What happens now? **Hint:** you may need to use **ctrl-C** to halt execution of you code.

j) Run your program in the debugger again. Where would you set a breakpoint to continue debugging the code?

k) In this case set a break point at **line 12**, run the program and print the contents of the data array. What are the contents of the data array?

l) How would you set a watch point on the index variable **i?** Do that now.

m) Continue running the code, using either the **next**, **step** or **continue** commands. What happens?

n) Looking at the output observed in the above step and the **line 13** of the **main.c** file, what is the problem with the execution of the code?

o) How would you correct the code on **line 13**? Make the correction, recompile, and run the code. **Congratulations, your code should now successfully run!**