GDB tutorial

Abhik Mondal

IIT Madras

August 18, 2014

GNU debugger

- Allows you to check what is going on inside the program during its execution.
- Generally used to figure out segmentation faults.

Compiling step

- One extra flag while compiling.

$$gcc \quad -g \quad < \mathit{file\_name} >$$

- This enables the built in debugging support required to support GDB.

Starting GDB

- The file to be loaded is the executable which is created on compiling.
- You can call

$$gdb\ < executable\_name >$$

- Look for the line

$$Reading\ symbols\ from\ < executable\_name >\ ...done.$$

Running Code

- '**run**' to start the code
- If there are no errors, it would say

$$'exited\ normally'$$

- A **seg fault** would give

$$Program\ received\ signal\ SIGSEGV,\ Segmentation\ fault.$$

Commands

- '**break** $<$ *lineno* $>$' to create a stop point in the code
- '**continue**' to resume, '**delete**' to remove breakpoints, '**next**' to execute next line, '**step**' to go line by line inside function, '**print** *varname*' to show variables, '**set args** $<$ *arg*1 $>$ $<$ *arg*2 $>$...' to input command line arguements
- You can use the first letter of the commands as short form.

References

- http://www.cs.umd.edu/ srhuang/teaching/cmsc212/gdb-tutorial-handout.pdf
- GDB Manual page (man pages) on the Unix system.

The End