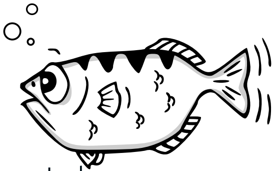
Intro to GDB



CS255 – Systems Programming Lab

John Malliotakis – jmal@csd.uoc.gr

Department of Computer Science, University of Crete, Heraklion, Greece

What is GDB?

GNU Debugger:

- Free and open source tool
- Lots of programming languages supported
- Debug your buggy code!
- Find out where your program crashes
- Pause execution before crash point.
- Examine/alter variables (prevent the bug!)

Setup and Usage

- GDB is already available on the department computers
- Must use **-g** flag when compiling your code (applies to *gcc* for C, *g++* for C++, *javac* for Java)
- Run with

  ```
  gdb <executable>
  ```
- Or with

  ```
  gdb --args <executable> <executable arguments>
  ```
- Or just

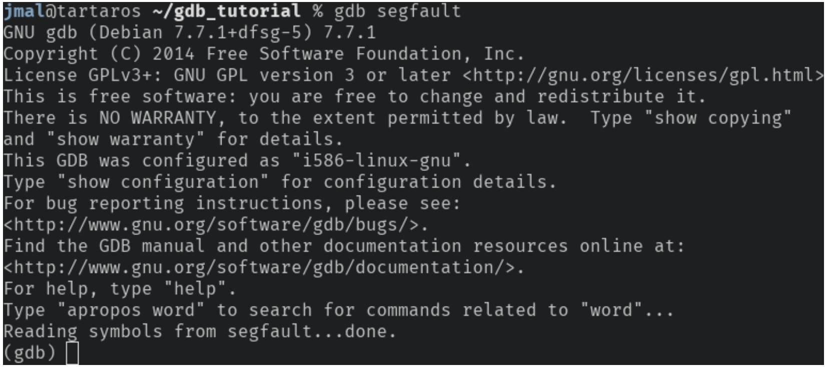
  ```
  gdb
  ```

  and specify an executable later

Functionality

gdb accepts commands from an interactive prompt

- Your program has not began executing yet
- The debugger is waiting for your command
- You should see something like this:



```
jmal@tartaros ~/gdb_tutorial % gdb segfault
GNU gdb (Debian 7.7.1+dfsg-5) 7.7.1
Copyright (C) 2014 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "i586-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from segfault...done.
(gdb) 
```

Basic Commands

Starting and Stopping

file	Choose an executable to debug
start	Begin execution and stop at the start of main
run	Begin execution without stopping at main, can specify args if not done already
quit(q)	Stop execution and quit debugger

Execution Flow Control

Our basic building block: the breakpoint

- Add (or remove) breakpoints at different parts of your code
- The debugger stops program execution at all breakpoints
- View the state of your program at the breakpoint

Commands for Execution Flow Control

Breakpoint Control

- **break(b)**: Instruct debugger to stop execution on specified point
    1. One of your function names
    2. **<File>:<Line>**: Specified line on specified file
- **delete(d) <Num>**: Remove breakpoint <Num> (or all breakpoints with no argument)
- **info breakpoints**: View all breakpoints

Commands for Execution Flow Control (Cont'd)

Execution Control

next(n)	Execute current (source language, not assembly!) instruction, go to next and stop
step	Like next, only different if source language instruction is a function call
until	Like next, only different regarding loops
continue(c)	Execute until hitting a breakpoint or the program finishes/crashes

Analyzing Runtime Info

We now know how to navigate our program

How do we actually inspect variables and function calls?

- **print(p)**: Our debugging swiss army knife
    1. Print any kind of variable
    2. print x: Display the current value for variable x
    3. Expressions in your source code language supported
    4. For example, print &x valid for a C program
    5. Can also change variable values, print x=1 actually changes x
- **backtrace(bt)**: Display function call stack (with function arguments)
    - Useful to understand the behaviour of our code and whether bad things can happen
- **frame(f)**: Display stack frame for current function
- **list(l) <linenumber/function>**: Print code around <linenumber/function>

More info about GDB, debugging

We only covered a tiny subset of the capabilities of GDB

- Use command **help** in gdb for a list of commands
- Use **help <command>** for more info on a specific command

GDB homepage https://www.gnu.org/software/gdb/

GDB documentation https://sourceware.org/gdb/current/onlinedocs/gdb/

GDB tutorial https://web.eecs.umich.edu/ sugih/pointers/summary.html

Questions?