M9 – Debuggers

CS 136L F23 – LEC 11

Yiqing Irene Huang, Qianqiu Zhang

UNIVERSITY OF WATERLOO | FACULTY OF MATHEMATICS
DAVID R. CHERITON SCHOOL OF COMPUTER SCIENCE

Disclaimer

- The following slides were not presented page by page in class.
- They are my own study notes to share with students.
- In the lab session, we will cover key points, do small demos and give hints on commonly seen errors

Main Points

Learn to use GDB debugger to debug C program

- Basic GDB commands to walk through the execution of a C program
- GDB commands to obtain program's internal state
- Navigating the C call stack
- Reversing the execution of statements one at a time
- GDB commands to trace through recursive function calls
- How to modify program's internal state to change its behavior
- Run GDB via the VS Code IDE

Lab Thresholds

Question	Description	# of Tests	Pass	Complete
Q1	Practicing GDB Commands	6	3	5
Q2	More GDB Practice	11	3	8

GDB

- You need to compile your code with `-g` option.
  - Also turn off the code optimization (i.e. use `-O0` option)
- GCC works better with GDB
- GDB acts on the executable
  - `gdb [executable_name]`
- GDB has its own commands
  - `help`
  - `run (r)`
  - `quit`
  - `layout src, tui enable/disable, Ctrl-X A`
  - `set style enable on/off`

GDB Basic Commands

Short	Full	Examples	Descriptions
r	run	`r` `r <t.in>t.expect`	Start execution of the program

b	break	`b main`
		`b 10`
		`b file.c:10`
			Set a breakpoint at the entry point of main function
			Set a breakpoint at line 10
			Set a breakpoint at line 10 of file.c

c	continue		Continue to the next breakpoint or the end of the program
i	Info	`i b`	List all breakpoints
d	delete	`d` `d 1`	Delete breakpoint(s)
en	enable	`en 1`	Enable breakpoint 1
dis	disable	`dis 2`	Disable breakpoint 2
n	next	`n`	Execute the next statement in the current context
s	step	`s`	Step into the function

GDB Basic Commands Cont'd

Short	Full	Examples	Descriptions

l	list	`l`
		`l 5`
		`l main`
			List N more lines after or around previous listing.
			List N lines around line 5
			List N lines around main function

p	print
	inspect
		`p var1`
		`p &var1`
		`p *ptr`
		`p var1+var2`
		`p arr@num`
		`p *arr@num`
			Print the value of var1
			Print the address of var1
			Print the dereference value of ptr
			Print the sum of var1 and var2
			Print addresses of num of elements in the array arr
			Print the values of num of elements in the array arr

set var	set var	`set var x=1`	Set the variable x value to 1 during debugging
whatis	whatis	`whatis var`	Print the type of the var

GDB Basic Commands Cont'd

Short	Full	Examples	Descriptions
disp	display	`disp var`	Print value of var each time the program stops
und	undisplay	`und 1` `info disp`	Cancel expressions to be displayed when program stops See code numbers of expressions to display when program stops
en disp	enable display	`en disp 1`	enable expressions with code number 1 to be displayed
dis disp	disable display	`dis disp 1`	disable expressions with code number 1 to be displayed
wa	watch	`wa var1`	Set a watchpoint for var1
del	delete	`d 3`	delete watchpoint with ID 3
en	enable	`en 3`	enable watch point with ID 3
dis	disable	`dis 3`	disable watch point with ID 3

Stack Navigation and Reverse Execution

The segmentation fault debugging

- Start the program (compiled with –g) with gdb
- Let it run till it generates a segmentation fault
- Use `backtrace` command to see the call stack
- Use `up` and `down` commands to move up and down the call stack
- You may do reverse-execution
  - Your will need to `record`
  - Then you can use `reverse-next` and `reverse-step` commands

GDB More Features

Short	Full	Examples	Descriptions
bt	backtrace where	`bt` `where`	print backtrace of all stack frames
up	up	`up`	move up the call stack
do	down	`down`	move down the call stack
set style	set style	`set style enable on` `set style enable off`	Enable CLI style Disable CLI style
set logging	Set logging	`set logging file debug.log` `set logging on`	Create a logging file Turn on the logging

GDB TUI Commands

Short	Full	Examples	Descriptions
layout	layout	`layout src`	Apply the src layout
CTRL-X A	CTRL-X A	`CTRL-X A`	Toggle TUI
i win	info win	`i win`	List of all displayed windows

fs	focus	`fs src`
		`fs prev`
		`fs next`
		`fs cmd`
			See focus to src window
			Set focus to previous window
			Set focus to next window
			Set focus to the command window

ref	refresh	`ref`	Refresh the terminal display

wh	winheight	`wh src 20`
		`wh src +2`
		`wh src -3`
			Set the src window height to 20 lines
			Increase src window the height by 2 lines
			decrease the height by 3 lines

Acknowledgement

- Slides by courtesy of Carmen Bruni
- Demo notes from Dave Tompkins
- Demo lectures by Carmen Bruni, Dave Tompkins, and Nomair Naeem

References

- CS 136L edX notes at https://online.cs.uwaterloo.ca/