

An Introduction to Debugging

An Introduction to Locating Segfaults with DDD

created by Stef Nychka

Department of Computing Science
University of Alberta

October 19, 2009

Overview

- 1 Background
- 2 Getting Started
- 3 Running Your Program
- 4 Locating Segfault
- 5 Summary and More Information

Background and Motivation

- a segfault (segmentation fault) occurs when a pointer refers to memory you can't access, and that pointer is dereferenced
- e.g.'s that *may* cause segfaults: overflowing an array (e.g. using an index that's too big), using an uninitialized pointer
- tough to find, but debuggers like ddd help
- ddd provides a more friendly graphical user interface to gdb (a text-based, but likely more stable debugger)
- This is just a brief introduction to using ddd to help with segfaults.

Getting Started

- download segfault.tar.gz from Exercise 5 (to a safe place.)
extract its contents.
- compile and run (`./avrg ./input`) to see segfault.
(don't need to understand code much for this presentation.)
- in SegFault dir., type `ddd ./avrg&` to run debugger and load avrg
- will get `.../init.c: no such file or directory. close ddd, and type make clean.`
- to fix, change Makefile so compiles with `-g`. must always do this. then type `ddd ./avrg&` again.

Running Your Program

- *Program* → *Run...* (*Run Program* window pops up)
- in *Run with Arguments*, type `./input` (for command line args.)
- click Run. note Segmentation Fault message at bottom.

Locating Segfault

- *Status* → *Backtrace...* (window pops up)
- top is caller, bottom callee
- click on lowest line referring to your code (*average* function)
- Note the segfault is happening in *fgets*, so one of its arguments must be causing the problem.
- could be *line* or *input*. it's *line*, and *line* should instead be an array.
 - note this type of error may not always result in a segfault
- *File* → *Exit*, then TA will fix code ... still seg faults, but is a different problem.

Summary and More Information

- compile with -g, ddd, run, backtrace
- What was shown today helps you locate the segfault. It does not deal with the actual cause.
- Note problems which cause segfaults at times do not result in a segfault ... it's still a problem that must be fixed.
- One more segfault in code. For Exercise (due by end of lab period), use ddd to locate it, and use C knowledge to fix problem.
- ddd can do much more. This is just an introduction.
- You can use valgrind, too. See Presentations link towards top of home page.
- Using preventative programming by initializing to 0 (NULL) helps.