More DDD

Breakpoints and Displaying Variables

created by Stef Nychka

Department of Computing Science
University of Alberta

November 4, 2009

Overview

- ddd does more than help with segfaults.
- You can jump to a particular line using breakpoints.
- You can execute a line at a time (stepping).
- You can view variables by displaying them.

- We will use these techniques to fix a problem in some code.

Noticing the Problem

- Download error.tar.gz from the Presentations page. The Presentations page is towards the top of the 201 page. Extract its contents. (It's similar to code from the 1st ddd pres.)
- cd into the Error directory, compile it (type `make`), then run it: `./avrg ./input`
- The average is one. Looking at the input file (use `cat`), this is not correct.

Brief Explanation of Code

You will need to understand a bit about the code.

- The function average puts integers from ./input into the `numbers` member:

```
struct vector_type {
  int max;
  int amount;
  int* numbers;
};
typedef struct vector_type vector_t;
```

  `storage` is the name of the vector_t variable in the average function.

- Note average calls `add` to put integers into `numbers`.
- The average of the integers is later determined by the compute function.
- Let's see if the problem is in the average function.

Setting a Breakpoint

- Type `ddd ./avrg &` to run ddd (and load avrg)
- Go to following line in the average function:

  ```
  init(&storage);
  ```

  (*Source* → *Display Line Numbers*, likely line 77.)
- Right click on line number, and choose *Set Breakpoint*
- *Program* → *Run*..., type `./input`, and click Run.
- Execution stops at the breakpoint.

Displaying Variables

- Display the numbers array by doing the following:
  - At the call to init, right-click on storage and choose *Display storage*
  - In the Data Window, where the contents of storage are shown, right-click on `numbers` and choose *New Display* → *Other*...
  - Change the Display Expression to `*storage.numbers@10`, and click the Display button.
  - Right click on border of *numbers (towards its edge), and choose *Rotate*.

  This displays the first 10 elements of the numbers array.
- You can hover over variables in the Source Window to see their current value, too, such as `line` and `number`.

Stepping Through the Code

- In the Command Tool (small, separate window with buttons), click Next to execute one line at a time.
- Do this until add is called a couple times.
- Note how `numbers`, `line` and `number` change.
- The number 1 keeps getting added to `numbers`, yet `line` and `number` have the correct values.
- ret_val is being passed to add. `number` should instead have been passed.

Click the Kill button in the Command Tool, fix the code, re-open program (*File* → *Open Program*...), and run again. You can do this later, on your own.

Summary

- Before running code, set breakpoint a bit before where you think the problem is.
- Can execute one line using Next (Step goes into functions, Next does not).
- View variables by displaying them
  - Can view vars. to help with a segfault, too.
- See http://www.gnu.org/software/ddd/manual/html_mono/ddd.html for more info.