Lab 8: Debuggers

Nathan Jarus

March 14, 2016

Introduction

This lab will give you experience using both `gdb` and `kdbg`. `git clone` the lab repository, then make a solutions file to write your answers in.

Problem 1: Segfaults

1. Compile 'list.cpp' and run it. Hopefully you get a segfault!
2. Open your executable in `gdb`.
3. Check the backtrace. What function is the segfault in?
4. Check that function (you can print out bits of code using the `list` command; try `list class::function_name`).
5. Set a breakpoint before the segfault, then `run` your code again and step through it.
6. As you are stepping through, inspect the value of `l`.
7. What is the bug in the code? Fix it.
8. Run your code and make sure you have fixed the segfault. You should have another problem now...

Problem 2: Loopy

1. At this point, you should have an infinite loop.
2. Open your code in `kdbg`. (Don't forget X-forwarding!)
3. Run the program, then select 'Break' from the 'Execution' menu.
4. Check the backtrace. What function is the infinite loop in?
5. Put a breakpoint on the loop, then restart the program.

6. Use the Locals explorer to expand the list a few levels.
7. 'run' the code a few times. What is happening to the addresses in the linked list?
8. Fix the code. (Hint: List has a copy constructor)
9. Run the fixed code. Hopefully it should terminate, but…

Problem 3: Math is hard

1. The sum is incorrect. Set a breakpoint that lets you watch what the sum function is doing.
2. Inspect the local variables to see what is going on.
3. Fix the code. Run it to make sure your fix worked.

Problem 4: Under the hood

What register holds the return value of functions like `List::value()`?

Epilogue

`git add` your answers and corrected code and `git push`!