Debugging Tips

James Wilcox and Kevin Zatloukal

May 2023

Debugging is the search from a *failure* seen by a user back to the *bug* that caused it. It is, in general, very difficult, often more-so than writing the code was initially.

Here are some tips to minimize the amount of debugging (searching) that will be required:

- **Write the simplest code that meets requirements.** "Debugging is twice as hard as writing the code in the first place. Therefore, if you write the code as cleverly as possible, you are, by definition, not smart enough to debug it." — Brian Kernighan
- **Practice defensive programming.** Write extra checks to make sure that callers pass valid arguments and that you return a valid result.

Here are some tips to keep in mind while debugging to help preserve your sanity:

- **Check the easy stuff first!** Make sure the file is saved, restart the server, etc. If the file is not saved, everything you do until saving is a waste.
- **Create a minimal test that demonstrates the bug.** It will be easier to debug this than the original failure containing lots of irrelevant details.
- **Look for silly mistakes** (e.g., typos). These are often the ones that slip past reasoning!
- **Make sure it is a bug.** Maybe it looks wrong, but in fact, that is what the spec says to do.
- **Write down what you have tried.** After debugging for a while, it is common to repeat searches of places you searched already. Avoid that by keeping track of what where you've searched.
- **Think of experiments that can shrink the search space.** Rather than just guessing exactly where the bug is, try to think of things you can try that will eliminate parts of the code as suspect.
- **Try explaining the problem to someone** (a.k.a. "rubber ducking"). Often, you'll figure it out while describing it.
- **Get some sleep.** The sleepier you are, the harder it is to see the bug. Often, after getting some rest, you'll spot the bug immediately.