CSE 374: Programming Concepts and Tools

Eric Mullen
Spring 2017
Lecture 12: gdb

Administrivia

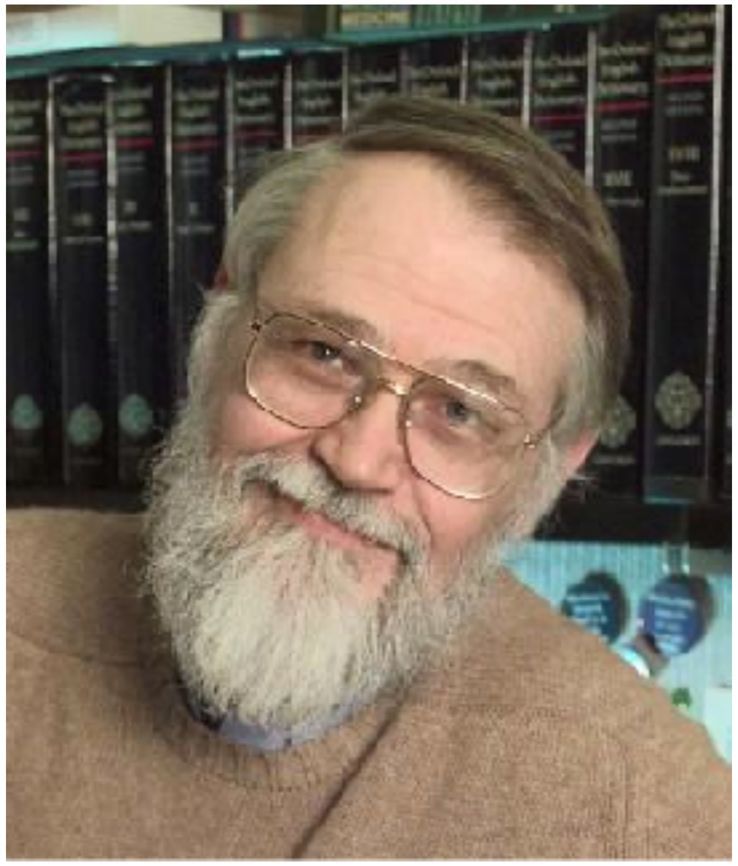
- HW3 due last night @midnight
- HW4 out now
  - First assignment in C!
  - Uses everything we've learned so far, up to today

Debugging

- How do you do it?
  - think
  - printf
  - read code, then think
  - use a debugger
  - think

Weirdly Appropriate Quotes

The most effective debugging tool is still careful thought, coupled with judiciously placed print statements.
-Brian Kernighan



Everyone knows that debugging is twice as hard as writing a program in the first place. So if you're as clever as you can be when you write it, how will you ever debug it?
-Brian Kernighan

Debugging Strategies

- Don’t put bugs in the program
- Think before typing: design before implementing
- Write down design in comments as you go
    - Anything Major: declaration+comments should be complete specification
    - Local Variables: name+comments should inform reader
- Keep comments up to date as you change program
- Turn on compiler warnings (-Wall -Wextra -Werror)
- Things can still go wrong…

More Help

- What if we had a ‘window’ into our running program?
- Why might that be helpful?
- A debugger is just this: it lets us stop and examine running programs

Debuggers

- Really useful tools, most languages have one
- `gdb` works for C and C++, java has its own
- Debugging can be hard and frustrating: this may help

gdb

- gdb (Gnu debugger) - part of standard linux toolchain
- gdb supports several languages (including C and C++)
- No fancy gui (unless you use emacs :P)
- Need to compile with -g to use
  - Otherwise info is very limited
- Running gdb: use `gdb <prog>` command
- When prompted: `run <args>`

gdb basics

- **r**un
- frame
- **p**rint
- info args
- info locals
- info break

- **b**reak <file:line>
- **b**reak <file:line> if <expr>
- **s**tep
- **n**ext
- **c**ontinue

A few tricks

- Everyone has their own

1. Always check why a segfault happens
2. Print pointer values to see how large something is
3. Stare at code even if it doesn’t crash
4. Print array contents (especially last element)

Advice

- Understand what the tool does and doesn't get you
  - gdb will solve some of your problems, but not all
  - thinking is still the best debugging tool
  - using it the first time will be slow, but the effort is worth it

Play with it!

The best way to learn gdb is to use gdb