COMP 2400 UNIX Tools

Christian Grothoff
christian@grothoff.org

http://grothoff.org/christian/

Debugging Strategy

- Have testcases to reproduce the error and limit scope
- Add assertions to narrow down possible causes: check the things that are easy to check first
- Use debugger to inspect call sequences and values to understand actual behavior
- Do **not** use the debugger to find problems
- Your brain is more important than the tool!

Debugging Task Flow

1. Observe your program. What does the program do for different inputs? Does it always do the same for identical inputs?

2. Formulate a theory. What kind of bug would explain the observed behavior?

3. Design and run experiments. What input would prove or disprove the theory? Where can the debugger help with the diagnosis?

4. Apply the fix. Does your fix resolves the original issue? Did it introduce new issues (regression tests)? Did the fox reveal other issues?

"Destillation" of Worst-Case Bugs

- Impossible to reproduce
- Non-deterministic
- Multiple issues interacting
- Hard to narrow scope

$\Rightarrow$ cut down application until bug "disappears"

gdb **Invocation**

- $ gdb binary-name
- $ gdb binary-name core-file
- Make sure binary is compiled with option -g
- Using -O0 (no optimizations) might also be useful

Using gdb

- (gdb) run ARGS
- (gdb) break FILENAME:LINE
- (gdb) bt DEPTH
- (gdb) s[tep]
- (gdb) n[ext]

Using gdb

- (gdb) info args
- (gdb) info locals
- (gdb) info threads

Printing and eXamining

- (gdb) print EXPRESSION
- (gdb) print array-ptr@size
- (gdb) x[/format] address
- (gdb) x/s a $\equiv$ (gdb) print (char*) a
- (gdb) x/NNNi main

Variables

- gdb automatically creates a variable ($NN) for any examined expression
- You can define your own using `set $NAME = EXPRESSION`

Creating Functions

- (gdb) define NAME
- > while x ¿ 50
- > step
- > end
- > print i
- > end

Arguments are $arg0, ..., $argN.

Executing Commands at Breakpoints

- (gdb) break filename.c:line
- (gdb) commands
- > silent
- > set x = 42
- > continue
- > end

Watchpoints

- (gdb) watch x – write only
- (gdb) rwatch x – read only
- (gdb) awatch x – read/write

Read watchpoints may only work with hardware support.

Remember

- The best way to eliminate bugs is to not write them
- The best debugger is your own brain
- Good testcases make debugging easier
- Not all bugs cause visible problems

$\Rightarrow$ Static analysis (next week!)

Questions

?