Using a Debugger

This is a video-focused exercise where I show you how to use the debugger that comes with your computer to debug your programs, detect errors, and even debug processes that are currently running. Please watch the accompanying video to learn more about this topic.

GDB Tricks

Here's a list of simple tricks you can do with GNU Debugger (GDB):

gdb --args Normally, gdb takes arguments you give it and assumes they are for itself. Using `--args` passes them to the program.

thread apply all bt Dump a backtrace for *all* threads. It's very useful.

gdb --batch --ex r --ex bt --ex q --args Run the program so that if it bombs, you get a backtrace.

GDB Quick Reference

The video is good for learning how to use a debugger, but you'll need to refer back to the commands as you work. Here is a quick reference to the GDB commands that I used in the video so you can use them later in the book:

run [args] Start your program with [args].

break [file:]function Set a break point at [file:]function. You can also use `b`.

backtrace Dump a backtrace of the current calling stack. Shorthand is `bt`.

print expr Print the value of `expr`. Shorthand is `p`.

continue Continue running the program. Shorthand is `c`.

next Next line, but step *over* function calls. Shorthand is `n`.

step Next line, but step *into* function calls. Shorthand is `s`.

quit Exit GDB.

help List the types of commands. You can then get help on the class of command as well as the command.

cd, pwd, make This is just like running these commands in your shell.

shell Quickly start a shell so you can do other things.

clear Clear a breakpoint.

info break, info watch Show information about breakpoints and watchpoints.

attach pid Attach to a running process so you can debug it.

detach Detach from the process.

list List out the next ten source lines. Add a – to list the previous ten lines.

LLDB Quick Reference

In OS X, you no longer have GDB and instead must use a similar program called LLDB Debugger (LLDB). The commands are almost the same, but here's a quick reference for LLDB:

run [args] Start your program with `[args]`.

breakpoint set --name [file:]function Set a break point at `[file:]function`. You can also use `b`, which is way easier.

thread backtrace Dump a backtrace of the current calling stack. Shorthand is `bt`.

print expr Print the value of expr. Shorthand is `p`.

continue Continue running the program. Shorthand is `c`.

next Next line, but step *over* function calls. Shorthand is `n`.

step Next line, but step *into* function calls. Shorthand is `s`.

quit Exit LLDB.

help List the types of commands. You can then get help on the class of command as well as the command itself.

cd, pwd, make just like running these commands in your shell.

shell Quickly start a shell so you can do other things.

clear Clear a breakpoint.

info break, info watch Show information about breakpoints and watchpoints.

attach -p pid Attach to a running process so you can debug it.

detach Detach from the process.

list List out the next ten source lines. Add a – to list the previous ten sources.

You can also search online for quick reference cards and tutorials for both GDB and LLDB.