ABOUT LLVM AND LLDB

This information is for students using OS X to compile and debug C programs.

Apple has moved permanently to LLVM, so it's probably just a matter of time before GDB stops working with the latest Xcode tools. Unfortunately, GDB is not anymore receiving Apple's support in order to read core dump files.

Anyway, we propose the following three options for debugging with OS X.

Option 1. You can use a Linux computer in the lab. GDB can read Linux core dumps.

Option 2. You can try using LLDB, http://lldb.llvm.org/index.html, which is the default debugger in Xcode on OS X. For those of you attempting to use LLDB this week to do the debugging, you can follow the following for Exercise 5:

```
% ulimit -c unlimited
% gcc -g -o div_zero div_zero.c
% ./div_zero
```

At this point, your program will crash and it will generate a core dump. The core dump will be located in the `/cores` directory. You need to check in that directory to see what the core file is called. Next, you read it with LLDB:

```
% lldb -c /cores/<core_file_name> div_zero
```

Now you are in with the core dump loaded, but you need to use LLDB commands (not GDB commands) to observe stuff. Please check at http://lldb.llvm.org/lldb-gdb.html for a list of commands and their GDB equivalents.

Option 3. You can install a different version from the stock gdb. According to this, you should be able to use gdb like Linux users to analyze OS X core dumps.

Installing MacPorts

- go to https://www.macports.org/install.php
- you should already have Xcode installed (if you have GCC). In that case, jump to installing MacPorts.
- In step (3) on the website, download and install MacPorts for your version of OS X.

Installing gdb-apple

Run the command:

```
% sudo port install gdb-apple
```

You can now use gdb-apple with the same commands as in gdb. This gdb-apple version can read OS X core dumps. For example, you can follow the same instructions for Exercise 5:

```
% ulimit -c unlimited
% gcc -g -o div_zero div_zero.c
% ./div_zero
```

At this point, your program will crash and it will generate a core dump. The core dump will be located in the `/cores` directory. You need to check in that directory to see what the core file is called. Next, you read it with gdb-apple:

```
% gdb-apple -c /cores/<core_file_name> div_zero
```

Code signing gdb

On some installations, gdb-apple must be granted certain permissions in order to be able to control other processes (i.e. control the process of the program being debugged). This can be done by "code signing" gdb. Here are a couple of sources with instructions on how to code sign gdb:

https://sourceware.org/gdb/wiki/BuildingOnDarwin
https://gcc.gnu.org/onlinedocs/gcc-4.8.1/gnat_ugn_unw/Codesigning-the-Debugger.html

Note that you need to replace "gdb" with "gdb-apple". Another possibility is to call gdb-apple with super user permissions: `% sudo gdb-apple`