

Using LLVM Sanitizers with Nim

Recently I've been using [Nim](#) as my go-to language for writing software in. One of the advantages that comes with Nim is the strong and easy to implement bridging behavior to C-family libraries. I recently put up a PR for Nim, which has since got merged, that enables some out of the box functionality for working with some of the Sanitizers that come as part of the LLVM toolchain.

The [Address Sanitizer \(ASAN\)](#) is a tool that enables runtime checking of memory behavior and usage. Likewise, the [Thread Sanitizer \(TSAN\)](#) is a tool to check for data races and other thread-safety issues. These are useful tools when working with code in any C-family language. While Nim isn't part of the C-family, the Nim compiler does convert Nim code into C before final compilation to assembly. Because of this, we can take advantage of the tooling provided to help debug our Nim code. To do this, a few additional flags need to be passed to the Nim compiler:

```
# set this if you have Xcode.app installed
clang_libraries_path="/Applications/Xcode.app/Contents/Developer/Toolch

# set this if you have the Xcode Command Line Tools installed
clang_libraries_path="/Library/Developer/CommandLineTools/usr/lib/clang

# this is to say use this search path for libraries to link against
--passL:"-L$clang_libraries_path"
# this is to say include the @rpath in the resulting binary to specify
--passL:"-rpath $clang_libraries_path"

# link against the ASAN dynamic library to take advantage of runtime ch
--passL:"-lclang_rt.asan_osx_dynamic"
# pass the flag to the C compiler to tell it to utilize the ASAN instru
--passC:"-fsanitize=address"

# link against the TSAN dynamic library to take advantage of runtime ch
--passL:"-lclang_rt.tsan_osx_dynamic"
# pass the flag to the C compiler to tell it to utilize the TSAN instru
--passC:"-fsanitize=thread"
```