

GCC support for the Darwin AArch64 ABI

Wednesday, 22 September 2021 08:00 (15 minutes)

This is a lightning talk.

One of the hurdles necessary to overcome for the M1 Darwin GCC port is supporting the Darwin ABI specification. GCC is designed to process argument passing the same way, regardless of whether the argument is named or variadic. This however does not leave scope to accommodate the Darwin modifications to the AArch64 ABI, which specifies that named stack-allocated arguments are passed naturally aligned, but variadic arguments are passed word-aligned.

To overcome this, we propose extending the GCC target hook API to carry the additional information necessary to let the backend make its own decisions about stack layout. The extension will not affect existing targets, and is opt-in by nature.

The second issue we tackled was support for the GCC nested function extensions to the C language. This is traditionally implemented using trampolines injected onto the stack at runtime, which requires an executable stack. Since Darwin's stack is not-executable, and the target doesn't make use of function descriptors, we required a solution to support nested function calls that didn't require changing the ABI.

Our preliminary plan is to generate the trampolines into an mmaped executable page: The trampolines will be generated when required within a function, and deallocated when the control leaves the enclosing scope.

I agree to abide by the anti-harassment policy

I agree

Primary authors: BLINOV, Maxim (Embecosm); BURGESS, Andrew (Embecosm); SANDOE, Iain

Presenters: BLINOV, Maxim (Embecosm); BURGESS, Andrew (Embecosm); SANDOE, Iain

Session Classification: GNU Tools Track

Track Classification: GNU Tools Track