

Link-time code generation invalidates a lot of classical assumptions about linking

 devblogs.microsoft.com/oldnewthing/20140606-00

June 6, 2014



Raymond Chen

The [discussion of DLL imports and exports](#) from a few years back relied heavily on what I called [the classical model of linking](#). Even though modern linkers can be quite non-classical, they do what they can to preserve certain aspects of classical behavior because many projects still rely on them.

Recall that the classical division of labor between the compiler and the linker is that the compiler takes source code and generates machine code with a few blank spaces that says, “Hey, linker, I’m not sure what number goes here, but I want it to be the address of *X*. Please patch the correct number in when you figure it out where *X* is.” The linker’s job is to go find *X*, assign it an address, and patch the address of *X* into all the places that the compiler asked for.

In the Visual Studio system, one of the ways of activating a large set of non-classical behaviors is to enable *Whole program optimization*, also known as *Link-time code generation*. When this is enabled, the division of labor between the compiler and linker shifts. The compiler still takes source code, but instead of generating machine code, it merely generates a parse tree (perhaps partially-digested). The linker then takes all these parse trees, combines them together (according to classical rules), and generates machine code.

Since the code generation is done with the full parse tree of the entire program, the code generator can perform very advanced operations, like [observing that a method is never overridden and can therefore be inlined](#). In particular, starting with Visual Studio 2012, the link-time code generator can even see that [you got `dllimport` wrong](#), and it goes back in time and redeclares the function correctly before proceeding with code generation.

Non-classical linking is even more advanced than non-classical physics. Whereas special relativity lets you stretch and slow down time, non-classical linking even gives you a limited form of time travel.

[Raymond Chen](#)

Follow

