What is DLL import hinting?

devblogs.microsoft.com/oldnewthing/20100317-00

March 17, 2010

Raymond Chen

Binding and *hinting* are two types of optimizations to improve the load-time performance of a module (executable or DLL). We'll start with hinting, then look at binding, and then look at how it affects delay-loading. The import table for a module contains a list of DLLs and a list of functions from that DLL which the module wishes to link to. The basic idea is that for each target DLL, the linker loads the DLL and then obtains the address of each imported function and from that DLL, records the results in the loaded module's table of imported function addresses. *Hinting* is a technique for speeding up this lookup. In addition to recording the name of the function the module wishes to link to, the linker also records the index of the function in the target DLL's export table. For example, suppose we had a DLL named `FLINT` whose export table looks like this:

1	`Barney`
2	`Fred`
3	`Wilma`

I've numbered the entries for reasons you'll see soon. You wrote a DLL which imports all three of these functions. The import table for your DLL goes something like this:

Import from `FLINT`	
	Fred (hint = 2)
	Wilma (hint = 3)
	Barney (hint = 1)

When your DLL gets loaded, the module loader loads the target DLL `FLINT.DLL`, and then it goes about resolving the imports. First up is `Fred`. But instead of just searching the export table of `FLINT.DLL` for the function `Fred`, it sees the hint and says, "Hey, let me look in slot 2 first." And lo and behold, there the function is in slot 2. Yay, a full search of the export table was not necessary; the hint sent us directly to the correct slot. The hint is just a hint, though. If `FLINT.DLL` doesn't have the function `Fred` in slot 2, then the loader just does things the old-fashioned way, by searching the export table for the `Fred` function. In general, hints are fairly resilient as long as the DLL doesn't change too much. If `FLINT.DLL` is updated, say by a security patch, the list of functions typically does not change, so the position in the exported names table remains unchanged. It's only if a function is added to or removed from `FLINT.DLL` do the hints begin to lose their effectiveness.

Next time, we'll look at *binding*, which is an optimization even more powerful than hinting, but which is also more fragile.

Raymond Chen

Follow