Why does the Windows Portable Executable (PE) format have both an import section and input directory?

devblogs.microsoft.com/oldnewthing/20231201-17

December 1, 2023

Raymond Chen

In the Windows Portable Executable (PE) file format, the import table directory entry contains the relative virtual address (RVA) of the import directory. But the imports table section (identified by the name `.idata`) also contains the relative virtual address of the import directory. Isn't this redundant? Why have two RVAs to the same thing? Can't we get rid of one of them?

The two entries correspond but they serve different purposes.

Sections describe where and how to allocate memory. They specify things like "I want to have 64KB of read-only memory at relative virtual address `0x00123000`, nd initialized with these contents." Directories describe what those various pieces of memory *mean*. "the import directory is at relative virtual address `0x00123000`."

For a meeting of an international committee, the representatives from each country are arranged at the meeting table, with the committee's current president at the head of the table. So why is there a placard in front of the president's seat that says "President"? Isn't that redundant?

Well, the president of the committee does, by convention and tradition, sit at the head of the meeting table, but there is no formal requirement that they sit there. Formally, what happens is that the committee members each have a seat, and the president of the committee is the person who also has a placard that says "President". The president may choose to sit somewhere other than the head of the table, and the placard lets everyone know who the president is.

There is no rule that says that the import table must begin at the start of a section named `.idata`, but that's how it is typically done, for reasions both traditional and practical.

The traditional reason is "That's how we've always done it."

One practical reason is that putting it in its own section makes it less likely to be sharing pages with data that is read-write, thereby increasing the likelihood that the pages end up shared when the module bindings are correct."

Bonus chatter: Of course, now that we have Address Space Layout Randomization (ASLR), module bindings are almost never correct, so that original reason is no longer effective. However, all of the import entries are likely to have the same address in all processes, so the logic still works, just in a different way: It is very likely that all the import entries *can be shared* among different processes since they are highly likely to be identical.