Why is the main() function always at address 0x00401000 in a simple program?

devblogs.microsoft.com/oldnewthing/20211006-00

October 6, 2021

Raymond Chen

If you compile a simple C or C++ program, and then load it into the debugger as a dump file (or if you execute the program with ASLR disabled), you'll find that the `main` function is at offset `0x00401000`. What is so special about this address?

It's the result of multiple technical decisions that add together, literally.

Your simple C or C++ program has only one function: `main`. It is therefore the function at the start of your code section, and the address of the `main` function is the address of the code section.

Traditionally, the code section is the first section of a Windows Portable Executable file. There's no technical reason for it, but *somebody* has to go first, and code seems to be the natural choice since it's almost always the most important part of the module. ("Primary reason for existence" in most cases.)

Sections are page-aligned because each section specifies its protection, and memory protection is applied at the page level. Therefore, the offset of the code section must be a multiple of the page size, which for x86 is 4KB.

The page at offset zero contains the module header information.

Therefore the first page available for the code section is the page at offset `0x1000`.

The last piece of the puzzle is that `0x00400000` is the default base address for executables on x86.

Put all of these decisions together (some technical, some arbitrary), and you find that the address of the `main()` function in a simple program is always `0x00401000`.

Raymond Chen

Follow