What does the N in nmake stand for?

devblogs.microsoft.com/oldnewthing/20190325-00

March 25, 2019

Raymond Chen

The Visual Studio toolchain comes with a tool called `nmake`. It processes files in roughly the same way as the traditional Unix `make` tool. Why is it called `nmake` instead of just `make`?

Rewind back to the late 1980's. The Microsoft languages toolchain[1] included a `make` program which we will generously describe as *vaguely inspired by* the Unix `make` tool: It processed files that looked like Makefiles, but its dependency calculations were nowhere near as sophisticated as the original Unix `make` program. As a result, you had to play games with the order of evaluation in order to get everything built in the correct order.

This woefully inadequate `make` program was the source of much dissatisfaction. A developer was selected to reimplement `make` from the ground up using a clean room design, working only from the makefile specification without any access to the Unix source code.

To distinguish the new version from the old and busted version, she gave the new version the name *New Make*, or `nmake`.

It's a tribute to the success of the new version that everybody has forgotten that there was an old and busted version to begin with.

Bonus chatter: What about `MASM`? What does the M stand for?

That one's easier. The M stands for Macro. It's the Microsoft Macro Assembler. What made it different from plain old `ASM` was the fact that it supported macros.

[1] The name *Visual Studio* wouldn't come onto the scene until a decade later.

Raymond Chen

Follow