

The old-fashioned theory on how processes exit

 devblogs.microsoft.com/oldnewthing/20070502-00

May 2, 2007



Raymond Chen

Life was simpler back in the old days. Back in the old days, processes were believed to be in control of their threads. You can see this in the “old fashioned” way of exiting a process, namely by exiting all the threads. This method works only if the process knows about all the threads running in it and can get each one to clean up when it’s time for the process to exit. In other words, the old-fashioned theory was that when a process wanted to exit, it would do something like this:

- Tell all its threads that the show is over.
- Wait for each of those threads to finish up.
- Exit the main thread (which therefore exits the process).

Of course, that was before the introduction of programming constructions that created threads that the main program didn’t know about and therefore had no control over. Things like the thread pool, RPC worker threads, and DLLs that create worker threads (something still not well-understood even today). The world today is very different. Next time, we’ll look at how this simple view of processes and threads affects the design of how processes exit.

Still, you learned enough today to be able to solve [this person’s problem](#).

[Raymond Chen](#)

Follow

