

Why do some process stay in Task Manager after they've been killed?

 devblogs.microsoft.com/oldnewthing/20040723-00

July 23, 2004



Raymond Chen

When a process ends (either of natural causes or due to something harsher like `TerminateProcess`), the user-mode part of the process is thrown away. But the kernel-mode part can't go away until all drivers are finished with the thread, too. For example, if a thread was in the middle of an I/O operation, the kernel signals to the driver responsible for the I/O that the operation should be cancelled. If the driver is well-behaved, it cleans up the bookkeeping for the incomplete I/O and releases the thread. If the driver is not as well-behaved (or if the hardware that the driver is managing is acting up), it may take a long time for it to clean up the incomplete I/O. During that time, the driver holds that thread (and therefore the process that the thread belongs to) hostage. (This is a simplification of what actually goes on. [Commenter Skywing gave a more precise explanation, for those who like more precise explanations.](#)) If you think your problem is a wedged driver, you can drop into the kernel debugger, find the process that is stuck and look at its threads to see why they aren't exiting. You can use the `!irp` debugger command to view any pending IRPs to see what device is not completing. After all the drivers have acknowledged the death of the process, the "meat" of the process finally goes away. All that remains is the "process object", which lingers until all handles to the process and all the threads in the process have been closed. (You did remember to `CloseHandle` the handles returned in `the PROCESS_INFORMATION` structure that you passed to `the CreateProcess function`, didn't you?)

In other words, if a process hangs around after you've terminated it, it's really dead, but its remnants will remain in the system until all drivers have cleaned up their process bookkeeping, and all open handles to the process have been closed.

[Raymond Chen](#)

Follow

