

---

Subject: Re: Program didn't exit in Task Manager after click [X]

Posted by [peterh](#) on Sat, 07 Nov 2020 11:01:57 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

So far I understand it, Windows will not terminate a thread forcefully, but will inject code to the thread, so the thread will terminate itself cleanly, when it runs this code.

This doesn't work, if the thread is IO waiting, because the thread doesn't run. In this case, if there is no IO timeout, the thread cannot self terminate.

So first the IO must be unblocked and ExitProcess does this for all IO in the process and threads. Also DLLs are detached cleanly from the threads.

I think, there is nothing wrong with this. Yes, it is system specific, but multithreading is always somewhat specific to the OS.

For console processes ExitProcess() is called automatically by Windows when Ctrl C is pressed, but not for GUI processes.

This is not a call from the debug API but from normal process API.

A cleaner way would be to set a termination flag which is polled by the threads, and I already tried this with no success.

The threads are blocked in some TCP Network call which obviously has no timeout. Maybe this can be improved, but I don't know.

I don't want to dig so deep into it.

---