A question about preventing the system from going to the idle state turns out to be misguided

devblogs.microsoft.com/oldnewthing/20141028-00

October 28, 2014

Raymond Chen

A customer asked how they could have their program prevent the system from going to the idle state. Specifically, when the system goes idle, the application gets into a weird state where it starts leaking memory like crazy. The program normally uses around 100MB of memory, but when the system goes idle, something funky happens and the program's memory usage shoots up to 4GB. To avoid this problem, they want to prevent the system from entering the idle state. Now, if your application is a special-purpose program running on a dedicated computer, then blocking the entry into the idle state might be acceptable. After all, the user bought the computer specifically to run your program and nothing else. But the description of the program provided by the customer did not suggest that this was the case. It was just some program being developed for a general audience. Interfering with the functioning of the entire system to hide a bug in your application is a horrible thing to do. It means that when your program is running, idle-time tasks never run, the computer never enters a low-power state, laptop batteries drain ten times faster than normal, and you basically ruin the entire computer. What you should do is debug your program and fix the memory leak.

This is like saying, "We manufacture car stereo systems, and we found that when the car is coasting, the power from the alternator is not sufficient to drive the speakers. We would like to prevent the car from coasting."

Raymond Chen

Follow