

ZOMG! This program is using 100% CPU!1! Think of the puppies!!11!!1!1!eleven

 devblogs.microsoft.com/oldnewthing/20101203-00

December 3, 2010



Raymond Chen

For some reason, people treat a program consuming 100% CPU as if it were unrepentantly running around kicking defenseless (and cute) puppies. Calm down already. I get the impression that people view the CPU usage column in Task Manager not as a diagnostic tool but as a way of counting how many puppies a program kicks per second.

While a program that consumes 100% CPU continuously (even when putatively idle) might legitimately be viewed as an unrepentant puppy-kicker, a program that consumes 100% CPU in pursuit of actually accomplishing something is hardly scorn-worthy; indeed it should be commended for efficiency!

Think of it this way: Imagine if your CPU usage never exceed 50%. You just overpaid for your computer; you're only using half of it. A task which could have been done in five minutes now takes ten. Your media player drops some frames out of your DVD playback, but that's okay, because your precious CPU meter never went all the way to the top. (Notice that the CPU meter does not turn red when CPU usage exceeds 80%. There is no "danger zone" here.)

Consider this comment where somebody described that they want their program to use less CPU but get the job done reasonably quickly. Why do you want it to use less CPU? The statement makes the implicit assumption that using less CPU is more important than getting work done as fast as possible.

You have a crowd of people at the bank and only ten tellers. If you let all the people into the lobby at once, well, then all the tellers will be busy—you will have 100% teller utilization. These people seem to think it would be better to keep all the customers waiting outside the bank and only let them into the lobby five at a time in order to keep teller utilization at 50%.

If it were done when 'tis done, then 'twere well / It were done quickly.

Rip off the band-aid.

Piss or get off the pot.

Just do it.

If you're going to go to the trouble of taking the CPU out of a low-power state, you may as well make full use of it. Otherwise, you're the person who buys a bottle of water, drinks half of it, then throws away the other half "because I'm thinking of the environment and reducing my water consumption." You're making the battery drain for double the usual length of time, halving the laptop's run time because you're trying to "conserve CPU."

If the task you are interested in is a low priority one, then set your thread priority to below-normal so it only consumes CPU time when there are no foreground tasks demanding CPU.

If you want your task to complete even when there are other foreground tasks active, then leave your task's priority at the normal level. Yes, this means that it will compete with other foreground tasks for CPU, but you just said that's what you want. If you want it to compete "but not too hard", you can sprinkle some `Sleep(0)` calls into your code to release your time slice before it naturally expires. If there are other foreground tasks, then you will let them run; if there aren't, then the `Sleep` will return immediately and your task will continue to run at full speed.

And cheerfully watch that CPU usage go all the way to 100% while your task is running. Just make sure it drops back to zero when your task is complete. You don't want to be a task which consumes 100% CPU even when there's nothing going on. That'd just be kicking puppies.

[Raymond is currently away; this message was pre-recorded.]

Clarification: Many people appear to be missing the point. So let's put it more simply: Suppose you have an algorithm that takes 5 CPU-seconds to complete. Should you use 100% CPU for 5 seconds or 50% CPU for 10 seconds? (Obviously, if you can refine your algorithm so it requires only 2 CPU-seconds, that's even better, but that's unrelated to the issue here.)

Raymond Chen

Follow

