

Don't try to allocate memory until there is only x% free

 devblogs.microsoft.com/oldnewthing/20120118-00

January 18, 2012



Raymond Chen

I have an ongoing conflict with my in-laws. Their concept of the correct amount of food to have in the refrigerator is “more than will comfortably fit.” Whenever they come to visit (which is quite often), they make sure to bring enough food so that my refrigerator bursts at the seams, with vegetables and eggs and other foodstuffs crammed into every available nook and cranny. If I’m lucky, the amount of food manages to get down to “only slightly overfull” before their next visit. And the problem isn’t restricted to the refrigerator. I once cleared out some space in the garage, only to find that they decided to use that space to store *more food*. (Who knows, maybe one day I will return from an errand to find that my parking space has been filled with *still more food* while I was gone.) Occasionally, a customer will ask for a way to design their program so it continues consuming RAM until there is only x% free. The idea is that their program should use RAM aggressively, while still leaving enough RAM available (x%) for other use. Unless you are designing a system where you are the only program running on the computer, this is a bad idea. Consider what happens if two programs try to be “good programs” and leave x% of RAM available for other purposes. Let’s call the programs Program 10 (which wants to keep 10% of the RAM free) Program 20 (which wants to keep 20% of the RAM free). For simplicity, let’s suppose that they are the only two programs on the system. Initially, the computer is not under memory pressure, so both programs can allocate all the memory they want without any hassle. But as time passes, the amount of free memory slowly decreases.

Program 10 (20%)	Free (60%)	Program 20 (20%)
Program 10 (30%)	Free (40%)	Program 20 (30%)
Program 10 (40%)	Free (20%)	Program 20 (40%)

And then we hit a critical point: The amount of free memory drops below 20%.

Program 10 (41%)	Free (18%)	Program 20 (41%)
-------------------------	-------------------	-------------------------

At this point, Program 20 backs off in order to restore the amount of free memory back to 20%.

Program 10 (41%)	Free (20%)	Program 20 (39%)
-------------------------	-------------------	-------------------------

Now, each time Program 10 and Program 20 think about allocating more memory, Program 20 will say “Nope, I can’t do that because it would send the amount of free memory below 20%.” On the other hand, Program 10 will happily allocate some more memory since it sees that there’s a whole 10% it can allocate before it needs to stop. And as soon as Program 10 allocates that memory, Program 20 will free some memory to bring the amount of free memory back up to 20%.

Program 10 (42%)	Free (19%)	Program 20 (39%)
Program 10 (42%)	Free (20%)	Program 20 (38%)
Program 10 (43%)	Free (19%)	Program 20 (38%)
Program 10 (43%)	Free (20%)	Program 20 (37%)
Program 10 (44%)	Free (19%)	Program 20 (37%)
Program 10 (44%)	Free (20%)	Program 20 (36%)

I think you see where this is going. Each time Program 10 allocates a little more memory, Program 20 frees the same amount of memory in order to get the total free memory back up to 20%. Eventually, we reach a situation like this:

Program 10 (75%)	Free (20%)	P20 (5%)
-------------------------	-------------------	-----------------

Program 20 is now curled up in the corner of the computer in a fetal position. Program 10 meanwhile continues allocating memory, and Program 20, having shrunk as much as it can, is forced to just sit there and whimper.

Program 10 (76%)	Free (19%)	P20 (5%)
Program 10 (77%)	Free (18%)	P20 (5%)
Program 10 (78%)	Free (17%)	P20 (5%)
Program 10 (79%)	Free (16%)	P20 (5%)

Program 10 (80%)	Free (15%)	P20 (5%)
Program 10 (81%)	Free (14%)	P20 (5%)
Program 10 (82%)	Free (13%)	P20 (5%)
Program 10 (83%)	Free (12%)	P20 (5%)
Program 10 (84%)	Free (11%)	P20 (5%)
Program 10 (85%)	Free (10%)	P20 (5%)

Finally, Program 10 stops allocating memory since it has reached its own personal limit of not allocating the last 10% of the computer's RAM. But it's too little too late. Program 20 has already been forced into the corner, thrashing its brains out trying to survive on only 5% of the computer's memory.

It's sort of like when people from two different cultures with different concepts of *personal space* have a face-to-face conversation. The person from the not-so-close culture will try to back away in order to preserve the necessary distance, while the person from the closer-is-better culture will move forward in order to close the gap. Eventually, the person from the not-so-close culture will end up with his back against the wall anxiously looking for an escape route.

Raymond Chen

Follow

