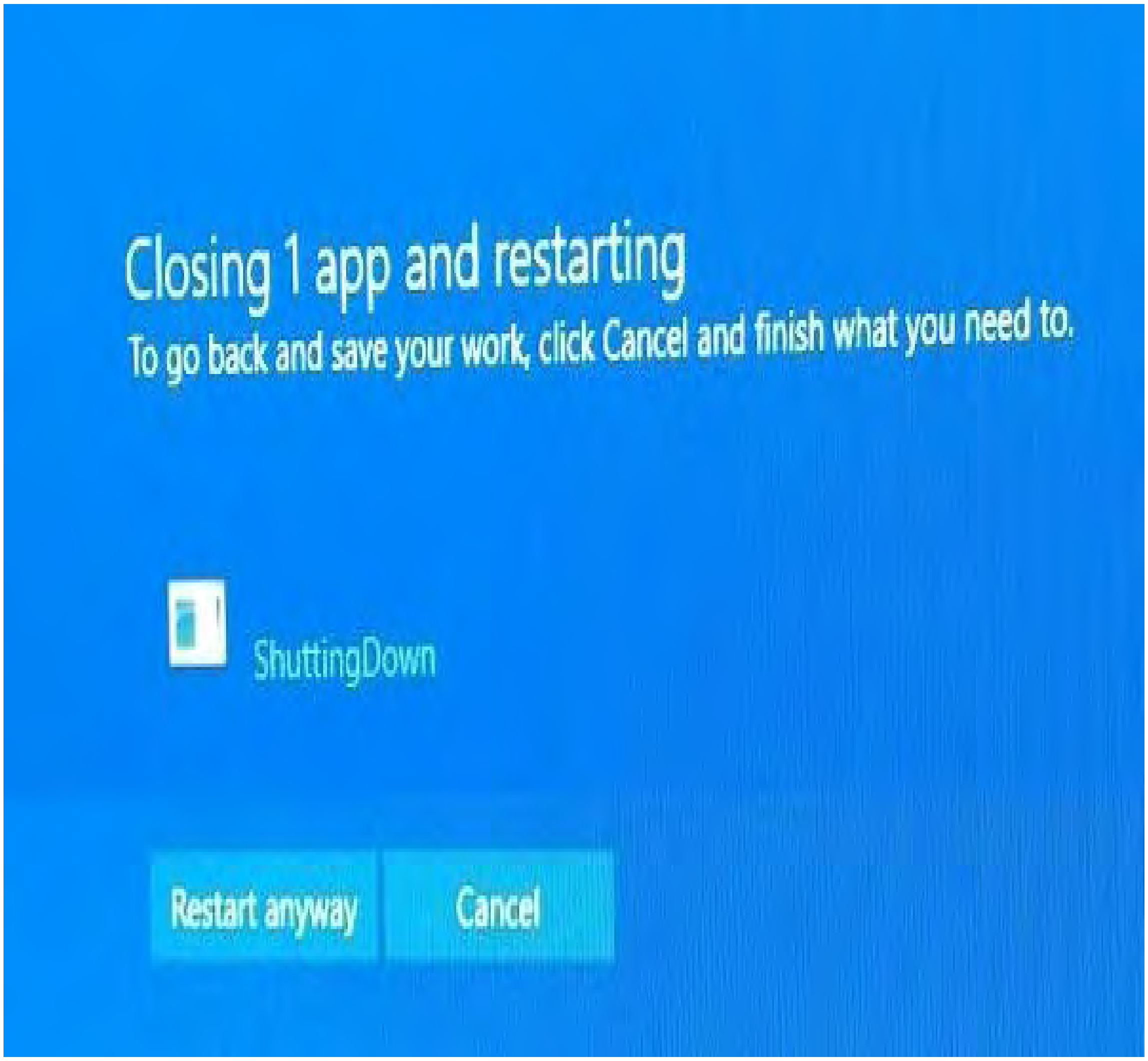
Science Translations
Established 1990
PC410.COM™

Managed Services,
PC Consulting, Sales, & Service in Central
Maryland

Why Does Windows Act Weird?

A lot of Windows weirdness is from a technology term called a ‘race event', like a speed race between lots of programs all trying to compete together. Basically, that's when software does a few things, and expects to get feedback, answers, or results from those things, and those result arrive in an unexpected order because other software is overwhelming the system. The program doesn't know what to do with that, and either triggers an error message, or locks up. Once a program locks up, it remains in memory, and Windows doesn't always know that has happened. And then you see this:



Closing 1 app and restarting

To go back and save your work, click Cancel and finish what you need to.

Translation: "The message that says that Windows is ‘ShuttingDown' isn't shutting itself down, so Windows can't shut down. Do you want to ignore "ShuttingDown" and shut down before Restarting, uh, Anyway?"

In the early days of personal computers, we had MS-DOS, the old Microsoft Disk Operating System. It could run one program and do one thing at a time. Multi-tasking (walking and chewing gum at the same time, or the computer equivalent), wasn't possible at first. So a program could do this:

```
Print 'hello '
Print 'world'
```

And that would reliably print ‘hello world' on some device, usually a monitor or a printer.

But somewhere around 1994, where we had the model 80386 processors and Windows 3.11, computer chips and software gained the ability to ‘time-share' so that programs could take turns running on the processor, and could do things without waiting for another program. That makes a few assumptions: All programs involved follow the rules, the operating system does everything right, and the hardware is functioning normally.

And if it doesn't, well, we might see ‘worldhello ‘ instead. That's a very basic failure of a race condition. Each line of code worked correctly, but the results are out of order. Far more likely: gibberish output and broken Windows.

Since then, we've gained multi-core processors that can run far more programs at the same time. We have ‘services' that run in the background, loading the Internet (communications), antivirus, system monitoring, video enhancements, surround sound, backups, phone-homes for both good and bad reasons, and on and on. As I write this, my computer shows 7 programs running, and 60 background processes (mostly services) either working or waiting for things to do. CPU load is at 2%, so this is normal operation for a modern PC with this year's specifications.

But what happens if we add a few more antivirus programs and have them monitor all incoming web pages and email? Well, they all try to do things at the same time, and things slow down dramatically. Add some more stuff, like browser toolbars, which mostly run all day long. Or view a web page that automatically plays videos when you arrive, maybe serveral animated advertisements and a full-motion video. At some point, the software overwhelms what the hardware can deliver, either too much for the processor to calculate or too much internet content to pull down. Pieces are skipped. Videos are time-sensitive; if the image pieces show up out of order, the image looks blocky or garbled. Error messages can occur if the programs involved don't know what to do with contents that are delayed, damaged, or out of order. Video programs know how to lower resolution or skip frames, so they can self-repair, to some extent.

Error messages, however, are written by humans, in advance. How do we do that? Well, the code is something like this, where lines 2 and 4 are ‘error traps':

```
Print 'hello ' (and listen for an error number)
If error number received is 6, display 'Printer wants paper.'
Print 'world'
If any error received, display 'Unknown error after printing.'
```

OK, so that works unless there is an error that wasn't expected at the time that it arrived. Most error messages say what the programmer thought was the obvious error that would happen after a specific line in a program. That's when something happens, not what. It's a guess.

Now multiply that by those 60 services and 7 programs that are running or waiting. Add some program code with incomplete error traps, or some software phoning home for advertising. The work area (memory) of the computer becomes messy and unstable, as programs receive web content or hardware status messages out of order, as gibberish, or not at all. In a severe case, the computer just locks up. But Windows does a fair job of cleaning up junk in memory, so what you see may simply be unexpected garbage.

So what does this mean for a computer user?

- Error messages tell you more about when something happened than what went wrong.
- Overloaded Windows gets weird.
- Programmers can't anticipate all delays or every possible error.
- Software doesn't deal with traffic jams well.
- Waiting for Windows to recover from an error works, sometimes. If you don't expect miracles.

And for things to do:

- Refresh can help a browser program like Firefox or Chrome when the page looks like junk because it didn't load completely. Press either F5, or Ctrl-R.
- Rebooting a computer clears memory and work areas. (Washes the blackboard.)
- Uninstall software that is no longer needed.
- Avoid installing software that runs all the time.

Contact

Address all editorial and unsubscribe requests to:
Jerry Stern, Editor, Science Translations, P.O. Box 1735, Westminster MD 21158

Phone (410) 871-2877