Historically, Windows didn't tend to provide functions for things you can already do yourself

devblogs.microsoft.com/oldnewthing/20100121-00

January 21, 2010

Raymond Chen

Back in the old days, programmers were assumed to be smart and hardworking. Windows didn't provide functions for things that programs could already do on their own. Windows worried about providing functionality for thing that programs *couldn't* do. That was the traditional separation of responsibilities in operating systems of that era. If you wanted somebody to help you with stuff you could in principle do yourself, you could use a runtime library or a programming framework. You know how to open files, read them, and write to them; therefore, you could write your own file copy function. You know how to walk a linked list; the operating system didn't provide a linked list management library. There are apparently some people who think that it's the job of an operating system to alleviate the need for implementing them yourself; actually that's the job of a programming framework or tools library. Windows doesn't come with a finite element analysis library either. You can muse all you want about how things would have been better if Windows had had an installer library built-in from the start or even blame Windows for having been released without one, but then again, the core unix operating system doesn't have an application installer library either. The unix kernel has functions for manipulating the file system and requesting memory from the operating system. Standards for installing applications didn't arrive until decades later. And even though such standards exist today (as they do in Windows), there's no law of physics preventing a vendor from writing their own installation program that doesn't adhere to those standards and which can do anything they want to the system during install. After all, at the end of the day, installing an application's files is just calling `creat` and `write` with the right arguments. Commenter Archangel remarks, "At least if the ACL route had been taken, the installers would have had to be fixed – and fixed they would have been, when the vendors realised they didn't run on XP." These arguments remind me of the infamous "Step 3: Profit" business plan of the Underpants Gnomes.

- Step 1: Require every Windows application to adhere to new rules or they won't run on the next version of Windows.
- ...
- Step 3: Windows is a successful operating system without applications which cause trouble when they break those rules.

It's that step 2 that's the killer. Because the unwritten step 2 is "All applications stop working until the vendors fix them."

Who's going to fix the the bill-printing system that a twelve-year-old kid wrote over a decade ago, but which you still use to run your business. (I'm not making this up.) What about that shareware program you downloaded three years ago? And it's not just software where the authors are no longer available. The authors may simply not have the resources to go back and update every single program that they released over the past twenty years. There are organizations with thousands of install scripts which are used to deploy their line-of-business applications. Even if they could fix ten scripts a day, it'd take them three years before they could even start thinking about upgrading to the next version of Windows. (And what about those 16-bit applications? Will they have to be rewritten as 32-bit applications? How long will that take? Is there even anybody still around who understands 16-bit Windows enough to be able to undertake the port?)

Raymond Chen

Follow