

Issue 080: Windows

De Programmatica Ipsum

2025-05-05

Contents

Issue 080: Windows	1
Why Wait For The Best When I Could Have You?	3
Dave Plummer	11
Raymond Chen	15

Issue 080: Windows



By Adrian Kosmaczewski, May 5th, 2025

Welcome to the 80th issue of *De Programmatica Ipsum*, about *Windows*.

In this edition:

- We deplore the sad state¹ of the Windows operating system.

¹<https://deprogrammaticaipsum.com/why-wait-for-the-best-when-i-could-have-you/>

- In the Library section², we review “The Old New Thing” by Raymond Chen³.
- In our Vidéothèque section⁴, we learn about how the Longhorn project derailed through Dave Plummer⁵.

Download this issue in DRM-free PDF⁶ or EPUB⁷ format, and read it on your preferred device.

We would like to thank our patrons who generously contribute every month (or have contributed in the past) to our work and help us run this magazine. Thank you so much! In alphabetical order: Adam Guest, Adrian Tineo Cabello, Benjamin Sheldon, Christopher Nascone, Colin Powell, Franz Lucien Moersdorf, Guillermo Ramos Álvarez, Jean-Paul de Vooght, Dr. Juande Santander-Vela, Patryk Matuszewski, Paul Hudson, Quico Moya, Roger Turner, Szymon Licau, and countless more leaving anonymous tips every month.

Enjoy this issue! Please subscribe to our free newsletter⁸ to stay updated about new releases, share the articles on social media, or contribute⁹ if you would like to support our work with a donation via Liberapay¹⁰.

Cover photo by Nikita Zaitsev¹¹ on Unsplash¹².

²<https://deprogrammaticaipsum.com/category/library/>

³<https://deprogrammaticaipsum.com/raymond-chen/>

⁴<https://deprogrammaticaipsum.com/category/videotheque/>

⁵<https://deprogrammaticaipsum.com/dave-plummer/>

⁶<https://deprogrammaticaipsum.com/pdf/issue-080-windows.pdf>

⁷<https://deprogrammaticaipsum.com/epub/issue-080-windows.epub>

⁸<https://deprogrammaticaipsum.com/newsletter/>

⁹<https://deprogrammaticaipsum.com/contribute/>

¹⁰<https://liberapay.com/akosma/donate>

¹¹https://unsplash.com/@necit15?utm_content=creditCopyText&utm_medium=referral&utm_source=unsplash

¹²https://unsplash.com/photos/white-and-blue-unk-unk-device-2Pg0P4pbEr8?utm_content=creditCopyText&utm_medium=referral&utm_source=unsplash

Why Wait For The Best When I Could Have You?



By Adrian Kosmaczewski, May 5th, 2025

Billions of people in this planet instinctively associate the concept of “operating system”

to Microsoft Windows by antonomasia¹³. Through perhaps resignation, corporate coercion, laziness, or just sheer ignorance, most cannot fathom a world in which laptops do not boot without a “Start” menu or do not randomly display a Blue Screen of Death every time they plug a new device¹⁴.

This is the reality of our world in 2025. Says who? Well, according to StatCounter¹⁵, at the time of this writing between 71% to 74% of all “desktop” computer users worldwide (that is, not counting smartphone and tablet users) are booting some version of Windows for their daily computing needs, while only 15% use macOS, and not even 5% use Linux.

These statistics, however, are vastly different among our cohort. According to Statista¹⁶,

In 2024, over 59 percent of developers worldwide reported using the Windows operating system for personal use, but only 48 percent reported using the OS when working. MacOS, based on Unix, had the second-largest share at 32 percent of developers using it for both personal and professional use, respectively.

These numbers are confirmed by (and, I suspect, might be even inspired by) those reported on the Stack Overflow 2024 Developer Survey¹⁷. The number of Linux users among developers jumps to more than 40%, all distributions considered.

But the gist of the latter developer survey is this gem: around 17% of participants declared using the Windows Subsystem for Linux, also known as WSL.

Ouch.

What happened to the famous “Developers, Developers, Developers”¹⁸ mantra of yesteryear? Well, turns out, a lot.

Let us go back in time 25 years. In June 2000 Joel Spolsky wrote a review¹⁹ of REAL-Basic, a cross-platform²⁰ development environment compatible with Microsoft Visual

¹³<https://deprogrammaticaipsum.com/antonomasia/>

¹⁴<https://www.youtube.com/watch?v=yeUyxjLhAxU>

¹⁵<https://gs.statcounter.com/os-market-share/desktop/worldwide>

¹⁶<https://www.statista.com/statistics/1482210/os-distribution-among-developers-worldwide/>

¹⁷<https://survey.stackoverflow.co/2024/technology>

¹⁸<https://www.youtube.com/watch?v=I-u8fo5esYI>

¹⁹<https://www.joelonsoftware.com/2000/06/12/realbasic/>

²⁰<https://deprogrammaticaipsum.com/issue-19-cross-platform/>

Basic²¹ and nowadays marketed with the horribly sounding name of Xojo²²:

Windows has 90% or 95% of the desktop market. Macintosh has 5% or 10%. If you're talking about office users, Windows is even more dominant; Macintosh has a bit more of the home users, but probably not even near 15%. What this means is that if you are a software developer, the only thing that makes sense financially is to develop a Windows version first.

How do things change!

Arguably, the only valid reasons to use Windows as a developer in 2025 are, first, to comply with corporate rules, and second, if you are developing apps for it, in which case you most probably use some combination of Visual Studio, C#, C++²³, and some other piece of Microsoft-sanctioned technology. Which, do not get me wrong: it is a terrific business, with an installed base of billions of potential customers for your product, not to mention the lucrative corporate market, an entire category in and of itself.

But, yes, that is as good as it gets. Vast numbers of developers are writing apps *without Windows* that run *without Windows*, and are very happy to ignore its mere existence. (Disclaimer: the author of these words falls in this category, precisely.)

Like for example “the Cloud”²⁴, with a capital C, where Linux reigns on production environments, and is also a prominent choice on the lap of developers. Or for Mobile²⁵, with a capital M, where, again, the Linux kernel and the Kotlin programming language power the most widely used category of devices, and where macOS and Swift are the technologies of choice (and necessity) for those writing iOS apps. In those two categories, Windows is effectively absent; or, if mandatory through corporate policy, then the WSL provides a refuge for productivity.

Honorable mention as well to all those Web²⁶ developers out there, these days usually referred to as “Frontend Engineers”, who would most probably settle for a Mac and an Adobe Creative Cloud subscription anytime.

Ask any developer working on Cloud, Mobile, or Web apps whether they would consider Windows a viable choice for their work, and their gasp would be heard from the

²¹<https://deprogrammaticaipsum.com/issue-59-basic/>

²²<https://en.wikipedia.org/wiki/Xojo>

²³<https://deprogrammaticaipsum.com/issue-77-cpp/>

²⁴<https://deprogrammaticaipsum.com/issue-22-the-cloud/>

²⁵<https://deprogrammaticaipsum.com/issue-44-mobile/>

²⁶<https://deprogrammaticaipsum.com/issue-52-the-world-wide-web/>

Pitcairn Islands. And let us not even start talking about LLMs, AI, or other golden nuggets *in vogue* these days.

(Yes, Horace, Unix has had its revenge²⁷.)

Even more damning is the statistical truth that the biggest fortunes in tech of the past 25 years (social media, cloud computing, gig economy, AI, mobile gaming, and more) have *not* been made through Windows apps. If anything, the *only* one who made insane amounts of money with Windows during the same period was, precisely, Microsoft.

When in doubt, follow the money.

Windows is celebrating a few milestones in 2025. Version 1.0 went RTM 40 years ago, in November 1985. Windows 3.0, released 35 years ago in May 1990, was the first commercially successful version, with ten million copies sold versus two million for Windows 2.

(If you are curious about those relics, you can try them online²⁸ for free nowadays.)

And of course, Windows 95 went on sale almost 30 years ago, in August 1995, with an orgasmic corporate feast fueled by dancing²⁹ to Rolling Stones' music, hysteria³⁰ in shopping centers, sketches³¹ by Jay Leno, and insane sales figures³².

But all of that awkward dancing on stage might not have happened at all. Unbeknownst to many, and as explained by Nina Kalinina³³, Windows 2 could have been the last ever version, were it not for the work of an engineer called David Weise³⁴. Larry Osterman told the world³⁵ what David did that was so groundbreaking for Microsoft³⁶ to crush IBM³⁷'s OS/2 once and for all:

You see, at this time, Microsoft's systems division was 100% focused on OS/2 1.1. All of the efforts of the systems division were totally invested

²⁷<https://www.asymco.com/2010/09/29/unixs-revenge/>

²⁸<https://www.pcjs.org/>

²⁹<https://www.youtube.com/watch?v=lAkuJXGldrM>

³⁰<https://rarehistoricalphotos.com/windows-95-launch-day-1995/>

³¹https://www.youtube.com/watch?v=_JzfROUDsK0

³²<https://techland.time.com/2013/05/07/a-brief-history-of-windows-sales-figures-1985-present/>

³³<https://www.ninakalinina.com/notes/win2/>

³⁴https://www.theregister.com/2005/02/05/windows_memory_man_leaves/

³⁵<https://learn.microsoft.com/en-us/archive/blogs/larryosterman/farewell-to-one-of-the-great-ones>

³⁶<https://deprogrammaticaipsum.com/issue-37-microsoft/>

³⁷<https://deprogrammaticaipsum.com/issue-62-ibm/>

in OS/2 development. (...)

And here was this little skunkworks project in building three that was sitting on what was clearly the most explosive product Microsoft had ever produced. (...) It ran Windows applications in protected mode, breaking the 640K memory barrier. (...)

There was just no comparison between the two platforms - if they had to compete head-to-head, Windows 3.0 would win hands down.

And win it did, all right³⁸. Peak Windows, as explained by Joel Spolsky above, happened around 2000; by 2005 the Longhorn and Vista debacle started eroding some of that success, but Windows is still here, still widely crashing, and still unbearable.

We are very far away from the “engineering odyssey”³⁹ that Dave Cutler⁴⁰, Lou Perazzoli⁴¹, and Mark Lucovsky⁴² undertook in the late 1980s. We are very far from the purported inspiration of Windows on the Unix design, something that Bill Gates himself openly stated at his Unix Expo keynote⁴³ in 1996:

And through Windows NT, you can see it throughout the design. In a weak sense, it is a form of Unix. There are so many of the design decisions that have been influenced by that environment. And that’s no accident. I mean, we knew that Unix operability would be very important and we knew that the largest body of programmers that we’d want to draw on in building Windows NT applications would certainly come from the Unix base.

Windows is not only bad⁴⁴, it is, was, and will be a train wreck of an operating system:

Microsoft released Windows XP on Oct. 25, 2001. That same day, in what may be a record, the company posted 18 megabytes of patches on its Web site: bug fixes, compatibility updates, and enhancements. Two patches

³⁸<https://archive.org/details/byte-magazine-1993-11>

³⁹[https://github.com/trimstray/technical-whitepapers/blob/master/base/Windows%20-%20A%20Software%20Engineering%20Odyssey%20\(Lucovsky\).pdf](https://github.com/trimstray/technical-whitepapers/blob/master/base/Windows%20-%20A%20Software%20Engineering%20Odyssey%20(Lucovsky).pdf)

⁴⁰https://en.wikipedia.org/wiki/Dave_Cutler

⁴¹https://en.wikipedia.org/wiki/Lou_Perazzoli

⁴²https://en.wikipedia.org/wiki/Mark_Lucovsky

⁴³<https://web.archive.org/web/20060208092929/https://www.microsoft.com/billgates/speeches/industry&tech/uexpo.asp>

⁴⁴<https://www.technologyreview.com/2002/07/01/40875/why-software-is-so-bad/>

fixed important security holes. Or rather, one of them did; the other patch didn't work.

Chances are that, as you read these lines, thousands of new employees are starting the first day at their new jobs, and as they sign their paperwork, they are all greeted by their IT department with a brand-new laptop (HP? Lenovo? Your guess is as good as mine), automatically bundled with Windows 11, Microsoft Office, and that new torture device called Microsoft Teams. Welcome to the company, kid.

The overall shitty Windows experience has not gotten any better in 30 years⁴⁵, and for those billions we keep talking about, this is precisely the normal state of things, no matter how much Mary-Jo Foley⁴⁶ or Paul Thurrott⁴⁷ rave about it.

These days Microsoft is more interested in shoving⁴⁸ something called "Recall"⁴⁹ down their users' throats, potentially threatening⁵⁰ the privacy and security of billions. They are showing ads on the Start Menu⁵¹, because why not. Satya gleefully states that at least 30% of all code produced by Microsoft is generated by AI⁵² as we speak.

And then, one day, CrowdStrike⁵³ happened. Microsoft quickly and rightfully offloaded all and any responsibility out of their legal system. After all, that is precisely what their EULA⁵⁴ explicitly said all along, have you read it?

Unfortunately, and thanks to the mighty powers of the Windows platform ecosystem⁵⁵, there is no other option than Windows for a large majority of those billion people. Be-

⁴⁵https://www.theregister.com/2023/12/19/windows_nt_30_years_on/?td=rt-3a

⁴⁶<https://web.archive.org/web/20100507185530/http://www.zdnet.com/blog/microsoft>

⁴⁷<https://www.thurrott.com/>

⁴⁸<https://arstechnica.com/security/2025/04/microsoft-is-putting-privacy-endangering-recall-back-into-windows-11/>

⁴⁹<https://gizmodo.com/windows-controversial-recall-is-back-heres-how-to-control-it-2000589002>

⁵⁰<https://arstechnica.com/security/2025/04/microsoft-is-putting-privacy-endangering-recall-back-into-windows-11/>

⁵¹<https://mastodon.social/@ghalfacree/111464902819204784>

⁵²<https://www.cnbc.com/2025/04/29/satya-nadella-says-as-much-as-30percent-of-microsoft-code-is-written-by-ai.html>

⁵³https://en.wikipedia.org/wiki/2024_CrowdStrike-related_IT_outages

⁵⁴<https://deprogrammaticaipsum.com/the-conquest-of-code/>

⁵⁵<https://deprogrammaticaipsum.com/geoffrey-g-parker-marshall-w-van-alstyne-sangeet-paul-choudary/>

cause, no, neither ReactOS⁵⁶ nor Free95⁵⁷ nor Wine⁵⁸ are entirely valid (or even usable) replacements for Windows at this point in time (seriously kids, stop the nonsense). This state of things is also the fault of all those other commercial vendors, including (but most prominently) IBM and Digital Research⁵⁹, who threw the towel as soon as Windows 95 hit the shelves, with the blissful complicit ignorance and ineptitude of a US government that knows nothing about technology⁶⁰.

Le sigh.

The lyrics of Lana Del Rey’s extraordinary 2019 song “Norman Fucking Rockwell” conjure veiled mockery, mind-numbness, and hypocrite praise, effectively conveying *l’air du temps* and a general feeling of mental fatigue in our society:

Goddamn, man-child
 You act like a kid even though you stand six foot two
 Self-loathing poet, resident Laurel Canyon, you know-it-all
 You talk to the walls when the party gets bored of you
 But I don’t get bored, I just see you through
 Why wait for the best when I could have you?

And that last one is, precisely, the phrase I have in mind every time I have to (reluctantly) boot a Windows laptop up.

Cover photo by Milad Fakurian⁶¹ on Unsplash⁶².

⁵⁶<https://reactos.org/>

⁵⁷https://www.theregister.com/2025/03/12/is_free95_real/

⁵⁸<https://www.winehq.org/>

⁵⁹<https://deprogrammaticaipsum.com/gary-kildall/>

⁶⁰<https://deprogrammaticaipsum.com/the-impossible-dialogue/>

⁶¹https://unsplash.com/@fakurian?utm_content=creditCopyText&utm_medium=referral&utm_source=unsplash

⁶²https://unsplash.com/photos/a-computer-screen-with-a-blue-screen-on-it-t_IkF_CNvSY?utm_content=creditCopyText&utm_medium=referral&utm_source=unsplash

Dave Plummer



By Adrian Kosmaczewski, May 5th, 2025

Hard to believe as it is, it has been already almost 20 years since the days when Apple plastered the halls of its Worldwide Developer Conference (WWDC) with huge posters⁶³ proclaiming “Redmond, start your photocopiers” or “Mac OS X Leopard: Hasta la vista, Vista”. The decade of the 2000s, coinciding with Steve Ballmer’s tenure as CEO, is widely perceived as a lost one for Microsoft, and one of the most visible signs of that decline was, without a doubt, the “Longhorn” saga⁶⁴.

⁶³<https://www.ifun.de/redmond-start-your-photocopiers-apples-wwdc-poster-im-der-letzten-11-jahre-40832/>

⁶⁴https://en.wikipedia.org/wiki/Development_of_Windows_Vista

Enter David William Plummer⁶⁵. He is a retired Microsoft engineer who worked in MS-DOS and Windows 95, and now runs the “Dave’s Garage”⁶⁶ channel on YouTube. Precisely, this month’s Vidéothèque movie is a recent entry⁶⁷ he published last March, where he dives into the nitty-gritty details of how, after the release of Windows XP, the company dived into a huge pool of hubris and mismanagement, collectively forgetting in the process how to release working software almost overnight.

Dave starts by pointing out that Longhorn is not exactly the same thing as Windows Vista, but it is one of the key milestones to understand one of the least well-received versions of Windows of all time. The core issue of Longhorn was, as it is the case for so many major software projects, the unbounded growth of its feature scope, or as it is commonly known, “scope creep”⁶⁸.

Among other examples, Dave explains how the new “WinFS”⁶⁹ flagship feature was going to provide a database system to search for files on the file system; something that, ironically enough, Apple quickly bundled into Mac OS X “Tiger” in 2005 as a feature called “Spotlight”⁷⁰, built with SQLite as its backend, and providing an incredible file search feature in the process. (Kids: before Spotlight, finding a particular file on a hard disk was almost impossible. I know, I know.)

At Redmond, however, to add insult to injury, the initial builds of WinFS were so buggy and ate so much RAM, that they would end up crashing machines... and beta users would end up disabling the feature altogether to be able to use Windows at all.

Ouch.

Longhorn would also be at the origin of the much touted “Aero”⁷¹ user interface style, which... well, was also mocked as a cheap rip-off of Mac OS X’s “Aqua”⁷² design idiom of the early 2000s. You see the “photocopier” pattern at play here.

Finally, according to Dave, one of the biggest roadblocks in the lifecycle of Longhorn was, of course, security. We have already⁷³ mentioned⁷⁴ Bill Gates’ “Trustworthy Com-

⁶⁵https://en.wikipedia.org/wiki/Dave_Plummer

⁶⁶<https://www.youtube.com/@DavesGarage>

⁶⁷<https://www.youtube.com/watch?v=RpRZ8BQiiMo>

⁶⁸https://en.wikipedia.org/wiki/Scope_creep

⁶⁹<https://en.wikipedia.org/wiki/WinFS>

⁷⁰[https://en.wikipedia.org/wiki/Spotlight_\(Apple\)](https://en.wikipedia.org/wiki/Spotlight_(Apple))

⁷¹https://en.wikipedia.org/wiki/Windows_Aero

⁷²[https://en.wikipedia.org/wiki/Aqua_\(user_interface\)](https://en.wikipedia.org/wiki/Aqua_(user_interface))

⁷³<https://deprogrammaticaipsum.com/microsofts-writings-on-security/>

⁷⁴<https://deprogrammaticaipsum.com/who-do-you-trust/>

puting” initiative and memo quite a few times in the pages of this magazine; needless to say, the pressure of fixing security bugs on Windows lengthened the development process quite a bit, and understandably so. And once Vista shipped... those same security features, involving checks and prompting the user for endless confirmations, made the first releases of the system strictly unusable, triggering an unprecedented backlash from press and users alike.

You get the idea. These were the days of the “Mini-Microsoft” blog⁷⁵, written by a (still) anonymous author. The days of Steven Sinofsky⁷⁶’s famous albeit scathing memo⁷⁷ about the state of Microsoft. The days of Ray Ozzie⁷⁸ sending the “Doomsday”⁷⁹ warning about the inevitability of the cloud. A troubled time for Microsoft until the arrival of Satya Nadella and the reorientation of the company away from the idea of Windows being a major cash cow for the organization.

Check this month’s Vidéothèque movie, “Windows Longhorn Explained by Dave Plummer”⁸⁰, on YouTube. His channel has other gems, like an interview of Dave Cutler⁸¹, the original chief architect of Windows NT, and also an explanation⁸² of the current monetization strategies used by Windows 11. Dave is also the author of the 2021 book “Secrets of the Autistic Millionaire”⁸³, a well-received work where he discussed his life with autism, ASD and Asperger’s, and how knowing what he knows now would have helped him in his younger years.

Cover snapshot chosen by the author.

⁷⁵<https://en.wikipedia.org/wiki/Mini-Microsoft>

⁷⁶https://en.wikipedia.org/wiki/Steven_Sinofsky

⁷⁷<https://hardcoresoftware.learningbyshipping.com/p/085-the-memo-part-1>

⁷⁸https://en.wikipedia.org/wiki/Ray_Ozzie

⁷⁹<https://www.computerworld.com/article/1540368/ozzie-s-doomsday-memo-warns-microsoft-of-post-pc-days-2.html>

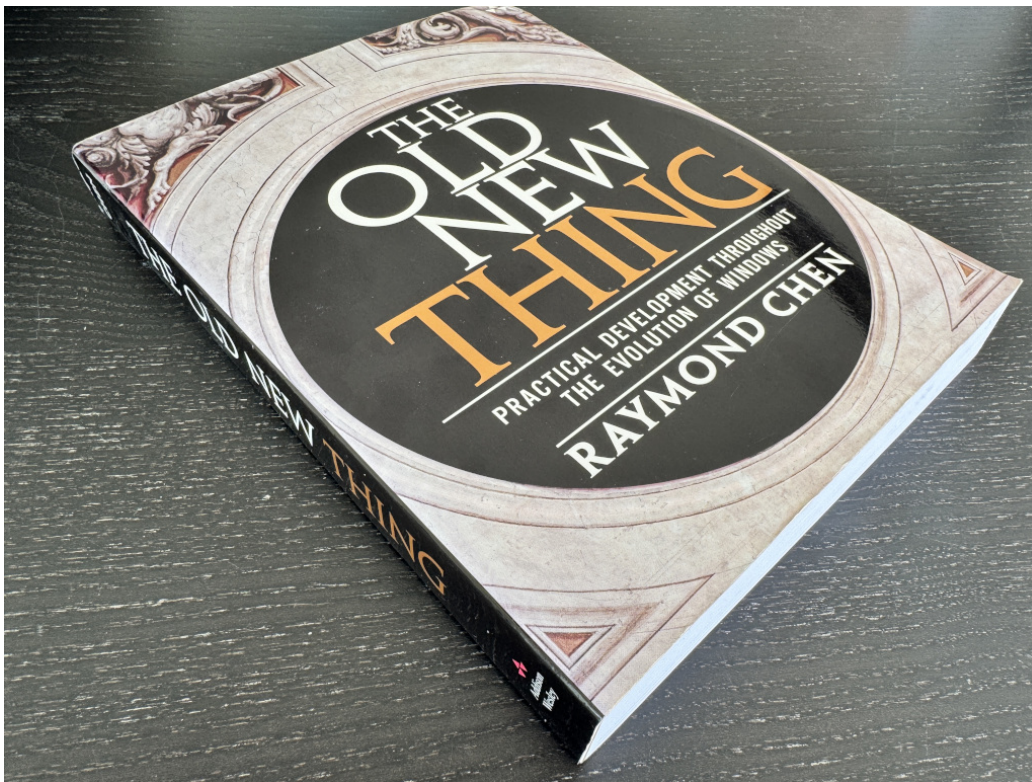
⁸⁰<https://www.youtube.com/watch?v=RpRZ8BQjiMo>

⁸¹<https://www.youtube.com/watch?v=xi1Lq79mLeE>

⁸²<https://www.youtube.com/watch?v=9KNhHThIU84>

⁸³https://www.reddit.com/r/autism/comments/qm90zx/new_book_secrets_of_the_autistic_millionaire/

Raymond Chen



By Adrian Kosmaczewski, May 5th, 2025

Most technical books I have read during my career have had a positive effect in me, in the sense that they brought me enthusiasm and eagerness to learn more, and to understand better the platforms, programming languages, and technologies I wanted to discover. This month's Library entry is probably the biggest exception to that rule; because as

soon as I read it, I decided to drop out from the Windows galaxy altogether, and to never look back.

Of course, this has nothing to do with the author's own storytelling capabilities, which are stellar no matter how you look at them. My reaction had more to do with the sense I got from opening the trunk of my Windows laptop and taking a look at the entangled mess of software wires that powered it. I did not like what I saw, and hence, I walked away, first becoming a die-hard iOS developer from 2008 to 2018, using macOS until Apple's "courage"⁸⁴ got the best of me, and finally jumping with both feet on the Linux bandwagon 7 years ago.

"The Old New Thing"⁸⁵ is a compilation of articles published in 2007, extracted from the blog of the same name, started around 2005 by Raymond Chen⁸⁶, who has worked in the Microsoft Windows team for literally decades. The blog still gets new entries every so often as this article hits the press.

This book should be, in the opinion of this author, a mandatory reading piece for whoever has to earn a living writing code for Microsoft Windows, in whatever layer of abstraction they might be working on, but in particular for those working on the lower levels of the building, close to the `basement` kernel.

As I said, the book is a delight to read; there are discussions about UI design; the weirdness of various Windows 95, 2000 and XP shenanigans; the problems of making pretty much every DOS program 100% compatible with later versions of Windows; a myriad references to the almighty Registry; the differences among various allocators (`SHGetMalloc` vs `SHAlloc` vs `CoGetMalloc` vs `CoTaskMemAlloc`, anyone? And you thought that `malloc`, `realloc`, and `calloc` were enough already); lots of crunchy details about COM object layout and behavior; and even sections explaining why files on Windows end with `CTRL+Z`, why they use "carriage return plus line feed" as a line separator, or why they cannot have filenames such as `CON` or `NUL`.

The index at the end provides quick access to dozens of articles explaining the inner workings of quite a few Win32 API calls (`CreateMenu()`, `GetVersion()`, etc.) and messages (`WM_PAINT`, `WM_MOUSEENTER`, etc.), with the corresponding source code snippets. This book is not the Windows equivalent of the 2010 book "The Linux Programming Interface"⁸⁷ by Michael Kerrisk, but feels more like watching a long documentary on

⁸⁴<https://akos.ma/blog/courage/>

⁸⁵<https://www.oreilly.com/library/view/the-old-new/0321440307/>

⁸⁶<https://devblogs.microsoft.com/oldnewthing/author/oldnewthing>

⁸⁷<https://www.man7.org/tlpi/>

National Geographic explaining the natural evolution and inner workings of a whole ecosystem.

To serve as an example, in chapter thirteen, “Backward Compatibility” (one of the most interesting of the book, by far) Raymond Chen explains how dealing with breaking software was a major priority for the team:

Why not just block the applications that rely on undocumented behavior?

Because every application that gets blocked is another reason for people not to upgrade to the next version of Windows. If you run the Registry Editor and look at HKEY_LOCAL_MACHINE\ SOFTWARE\ Microsoft\ WindowsNT\ CurrentVersion\ Compatibility, you’ll see a list of programs that would have stopped working when you upgraded from Windows 3.0 to Windows 3.1.

(I do not have a Windows machine at hand, so I will leave it as an exercise to the reader to verify that this is still the case in Windows 11.) Later in the same chapter, a visibly adamant author explains that

The LOB (line-of-business) application is the deal-breaker. If a Windows upgrade breaks a LOB application, it’s game over. No upgrade. No company is going to lose a program that is **critical to their business**.

(Emphasis in the original.)

This month’s Library book, in the sense of the excerpts shared above, is a concrete explanation of the various ways in which Microsoft *embraced, extended, and extinguished*⁸⁸ the market for alternative PC operating systems starting in the 1990s. We owe the current state of the world of computers, to a large degree, to the stubbornness of the Windows team, who made sure that *every single DOS and Windows app under the sun* would work (or at least, launch) in later versions of the operating system. A certainly Herculean task, one that no other vendor ever attempted; but one that cost users and developers their sanity in the long run.

In terms of style, the book transpires the exasperation and contempt directed towards third party creators of Win32 apps, drivers, and components, who would bypass the APIs and rules put forward by the Windows team, watch their software crash miserably, and then complain to Microsoft, only to find the wrath of Raymond on the other side of the line. Examples abound all along the book, and they are seriously delightful.

⁸⁸https://en.wikipedia.org/wiki/Embrace,_extend,_and_extinguish

The comments in this article, of course, should not detract from the notion of the insane amount of work that was required to make a single operating system work with the huge variety of software and hardware existing in the IBM PC ecosystem. From a purely technical and project management perspective, Windows was, and always will be, a groundbreaking and extraordinary achievement, held by a team of incredibly smart people who did their best. The point of this article is, rather, that human society in the 21st century, as a whole, might probably have been better off *without* Windows crashing around at every corner.

After its lecture, this whole book begs the question of the overall sanity and coherence of the Windows operating system, and quite simply provides a direct answer to the perceived brittleness of the ensemble since, well, always. To put it shortly, if you are person in need of actual reasons *not* to use Windows, this is definitely the book for you.

(The problem is, however, that most *users* of Windows will never read, let alone hear about this gem of a book, and that is a pity.)

Cover photo by the author.