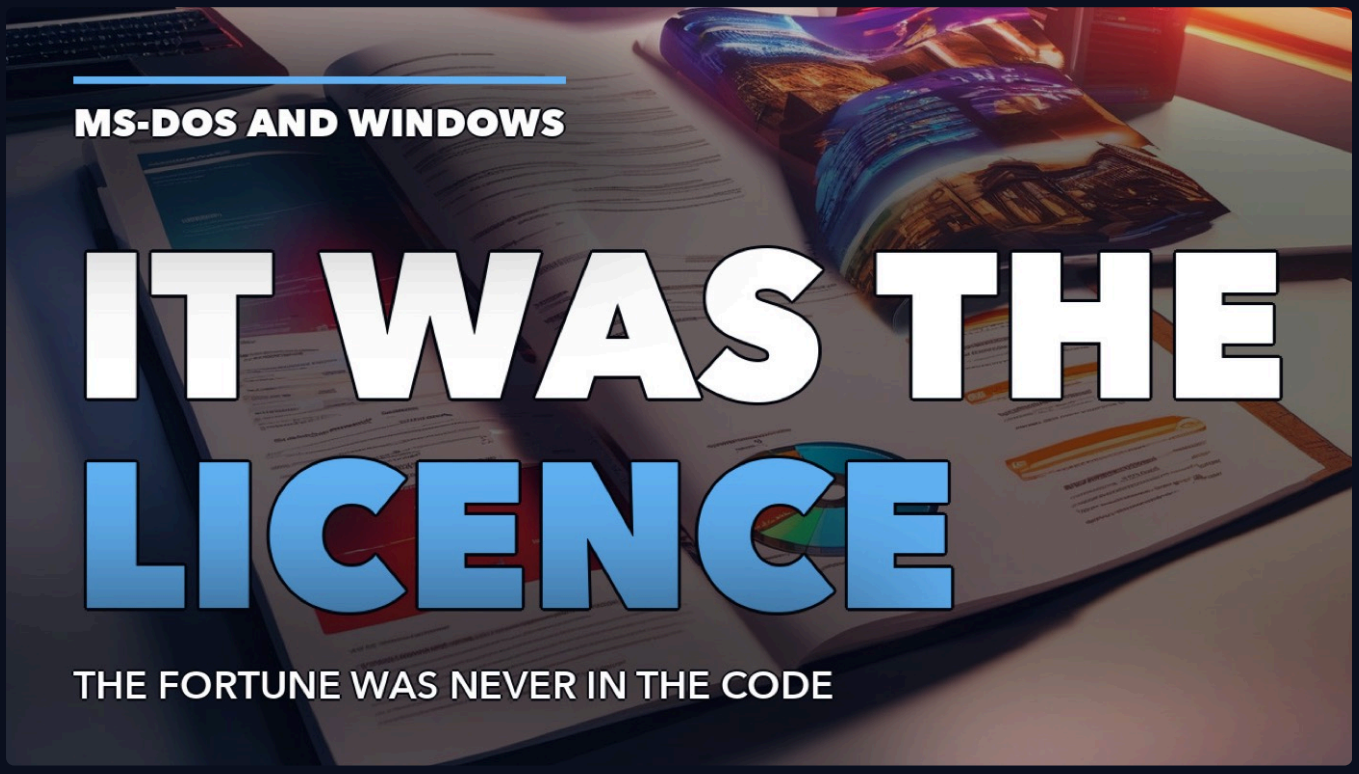
Forwards or Backwards

https://forwardorbackwards.com



Microsoft: DOS and Windows, a Timeline

Watch on YouTube: https://www.youtube.com/watch?v=4yaxP6b5ae0

The operating system that built Microsoft was not written by Microsoft. It was written by Tim Paterson at Seattle Computer Products, and Microsoft acquired it in two payments: a flat fee of USD 25,000 in December 1980, then all rights for an additional USD 50,000 in the summer of 1981. USD 75,000 in total.

The engineering was competent and unremarkable. What was remarkable was a clause: Microsoft retained the rights and licensed the system to other computer manufacturers, calling it MS-DOS.

And because IBM had built its personal computer out of parts anybody could buy and published the circuit designs, there turned out to be an enormous number of other manufacturers. IBM made the market open. Microsoft owned the one part of it that was not.

That is the first thing this film is about: the decisive asset was contractual, not technical.

The second is a pattern. Once you own the operating system, the threat is never another operating system. It is whatever sits on top of yours and makes it interchangeable — a layer above your layer, that the user deals with instead of you. That threat has arrived four times.

The first time, Microsoft built the layer itself: Windows, shipped 1985, running on top of DOS, and by version three in 1990 the relationship had inverted — Windows was what you bought and DOS was the plumbing underneath. A company that replaced its own primary product while owning both ends.

The second time, somebody else built it. The browser ran on any operating system, which would have made Windows interchangeable, and Microsoft's own word for the danger was middleware. It built a competing browser, gave it away, and integrated it into Windows. Commercially that worked. Legally, on 18 May 1998, the United States filed suit.

This film gives all four stages of that case, because any one quoted alone is false to the record: conclusions of law on 3 April 2000 (monopoly illegally maintained); the remedy on 7 June 2000 (separation into two companies); the Court of Appeals on 28 June 2001, which affirmed the monopoly-maintenance finding, reversed attempted monopolisation, remanded the tying claim and vacated the breakup; and the revised proposed final judgment of 6 November 2001, with no restructuring. Microsoft was not broken up. Microsoft did not win.

The third time, the layer left the machine altogether and went into a pocket — and the same mechanism that had made Windows untouchable on the desk excluded Microsoft from the phone. This chapter carries no market-share figure, because none was traced to a primary source.

And the accounts say why. In the quarter reported 29 July 2026, Intelligent Cloud was USD 39.306 billion, up 32 per cent, with Azure up 43. More Personal Computing, the segment containing Windows, was USD 12.854 billion — down 4 per cent, the smallest of the three and the only one shrinking.

Chapters

1. The fortune was in the licence, not the code
2. A contract for software they did not have
3. Twenty-five thousand dollars, then fifty thousand more
4. The clause
5. The people who sold it
6. A layer on top of their own layer
7. The version that worked
8. The twenty-fourth of August, 1995
9. The layer above the layer
10. The eighteenth of May, 1998
11. What the appeals court actually decided

12. The layer moves into a pocket
13. Renting the machine instead of selling it
14. What the numbers say now
15. The second of June, 2026
16. Forwards or backwards

1. The fortune was in the licence, not the code



There is a version of this story where a company writes a brilliant piece of software and the world beats a path to its door.

That is not what happened, and the real version is more interesting.

The operating system that built Microsoft was not written by Microsoft. It was written by somebody else, at another company, and Microsoft acquired it in two payments totalling seventy-five thousand dollars. The engineering was competent and unremarkable. What was remarkable was a clause.

Microsoft kept the right to sell that operating system to everybody — not only to the customer who had asked for it. And because of a decision made at that customer, which we will come to, there turned out to be an enormous number of everybody.

So the first thing this video is about is that the decisive asset was contractual rather than technical. Microsoft did not need to out-engineer a rival operating system — it needed the right written into the deal for the one it had acquired. That clause is what the rest of this video traces.

The second thing is what happened next, and it is a pattern rather than an event. Once you own the operating system, the threat to you is never another operating system.

A layer above your layer, that the user deals with instead of you — that is the shape the threat always takes. It sits on top of what you built and makes your part of the machine interchangeable, however good your code is. Everything that follows in this video is a version of that same shape arriving again.

That threat has arrived four times in this story, in four different costumes, and Microsoft's response each time tells you more about the company than any product launch does. Watch for the same question each time it returns: does Microsoft own the new layer, or does somebody else?

We start in November nineteen eighty, with a contract for something that did not exist.

2. A contract for software they did not have



In November nineteen eighty, Microsoft signed a contract with IBM to supply an operating system for a personal computer IBM was building.

The Computer History Museum, which now holds and has published the original source code, puts the position plainly: when Microsoft signed that contract, they had no such operating system. It is a striking admission for a museum to make in its own name, because it means the product came after the sale, not before it. What filled the gap is the subject of the next two chapters.

This is worth sitting with, because it inverts the usual order. The sale came first. The product was acquired afterwards, to fill an obligation already entered into.

What Microsoft did have was a relationship and a reputation for delivering programming languages. Decisively, it also had a willingness to say yes to a deadline that a more cautious supplier would have declined.

Nearby, at a company called Seattle Computer Products, an engineer named Tim Paterson had written an operating system for the same family of processors IBM had chosen. It was a practical, unglamorous piece of work, built because his employer needed something to run on the hardware they were selling. He had no reason to think it would end up anywhere near IBM.

It was not designed to be the foundation of anything. It was designed to work by a deadline, which is a different specification entirely, and in this case the right one.

In December nineteen eighty, Microsoft licensed it. And that licence, rather than the code inside it, is the beginning of everything that follows. The next chapter puts an exact price on that licence, and on what came after it.

3. Twenty-five thousand dollars, then fifty thousand more



The numbers here are small enough to be worth saying slowly, because they are the most consequential figures in the video and they are almost always reported wrongly.

In December nineteen eighty, Microsoft licensed the operating system from Seattle Computer Products for a flat fee of twenty-five thousand dollars. That is a licence, not a purchase — permission to use something on terms, while Seattle Computer Products continued to own it outright. The purchase, when it came, cost more.

Then, in the Computer History Museum's words, by the next summer they recognised the importance of owning it — and they purchased all rights for an additional fifty thousand dollars.

Two transactions. Twenty-five thousand in December nineteen eighty. Fifty thousand more in the summer of nineteen eighty-one.

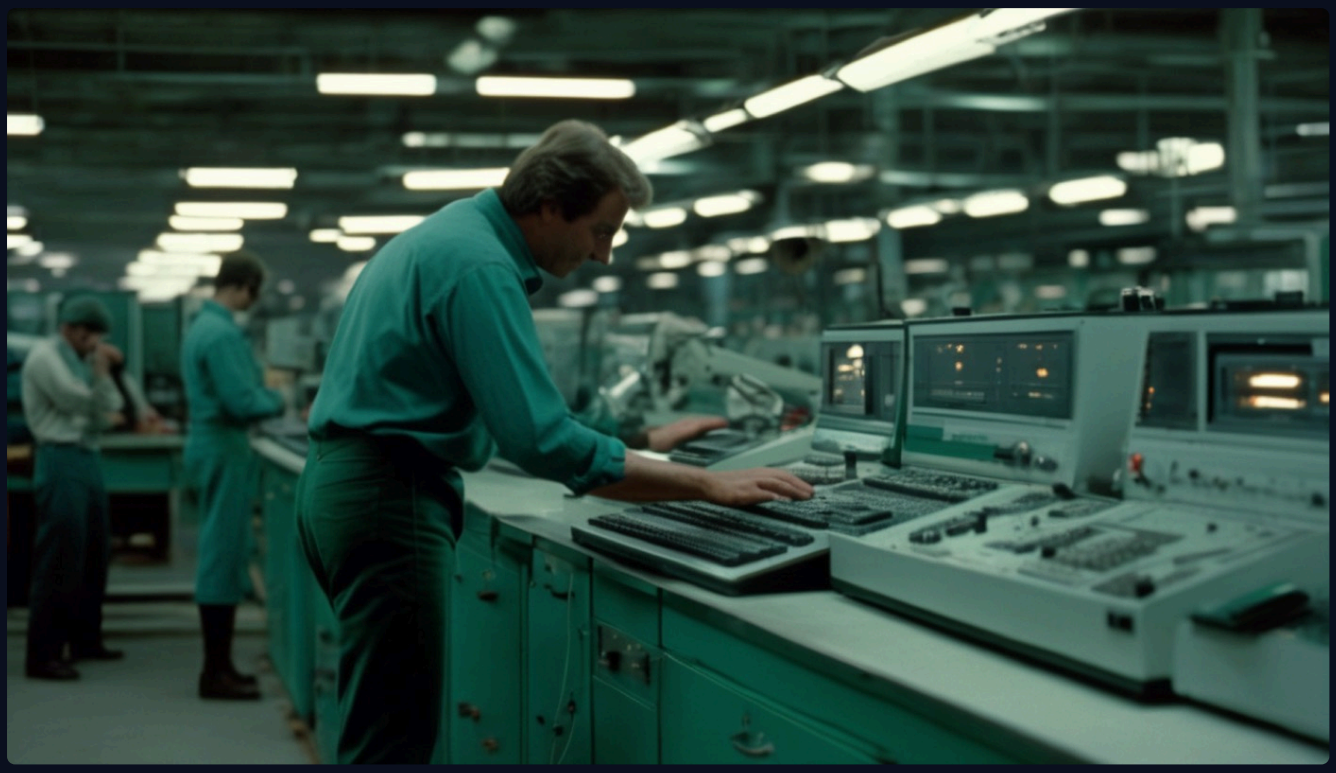
Seventy-five thousand dollars in total, and I am giving you the total only after both parts. The figure you will usually see quoted is the fifty thousand on its own, which is the second payment and not the price. Both payments together are what actually bought the software outright.

Tim Paterson left Seattle Computer Products to work for Microsoft in nineteen eighty-one, and spent eleven months there working intensively with the IBM engineers. The man who had written the software by a deadline was now the one finishing it against IBM's own deadline. That overlap is why the code and the company's fortunes end up so tightly bound together.

When IBM released its personal computer in August nineteen eighty-one, it shipped with that software, under IBM's name.

And the difference between the licence and the purchase is the entire company. A licensee sells what it is permitted to sell. An owner sells to anyone it likes.

4. The clause



Here is the sentence that matters, and it comes from the museum's own account: Microsoft retained the rights to the operating system, and licensed it to other computer manufacturers, calling it MS-DOS. IBM had bought the right to use it on its own machine. Everyone else who wanted it had to go to Microsoft, not to IBM.

Read that against what IBM had done with the machine itself.

IBM built its personal computer out of parts anybody could buy, and then published the technical reference — the circuit designs, the source code for the built-in software. That decision made the machine an industry standard, and it is treated at length in the companion video on IBM.

The consequence for this story is direct. Because the hardware was reproducible, other manufacturers built compatible machines. Because those machines were compatible, they needed the same operating system. And because Microsoft owned that operating system rather than merely supplying it to IBM, every single one of those manufacturers came to Microsoft.

IBM had made the market open. Microsoft owned the one part of it that was not. Every clone maker who wanted to build an IBM-compatible machine still had to come to Microsoft for the one piece IBM had not published.

I want to be careful not to make this sound like a trick. It was a negotiated term, agreed by a party that had its own reasons, at a moment when nobody in the room knew what the personal computer was going to become. Very little about it looked clever at the time.

But it is the reason that a company which had not written an operating system, and had paid seventy-five thousand dollars for one, ended up collecting from an entire industry. Every manufacturer that wanted to compete with IBM had to pay the same company IBM itself had once paid. That is the clause working exactly as it was written to.

5. The people who sold it



There is a part of this story that tends to get left out, and leaving it out makes the whole thing sound cleaner than it was.

Seattle Computer Products sued.

The company that had written the software, and sold first a licence and then all rights, went to court against the company that had turned it into the foundation of the personal computer industry. The Computer History Museum records the outcome in one line: they ultimately settled out of court for almost one million dollars. That is the number the museum states, and it is the one this video uses.

Now, what does that establish, and what does it not?

It establishes that there was a genuine dispute about what had been sold and on what terms, that it was serious enough to litigate, and that it ended in a payment of roughly thirteen times the original seventy-five thousand. That is the scale of the dispute — a payment worth more than ten times what Microsoft had first paid for the software.

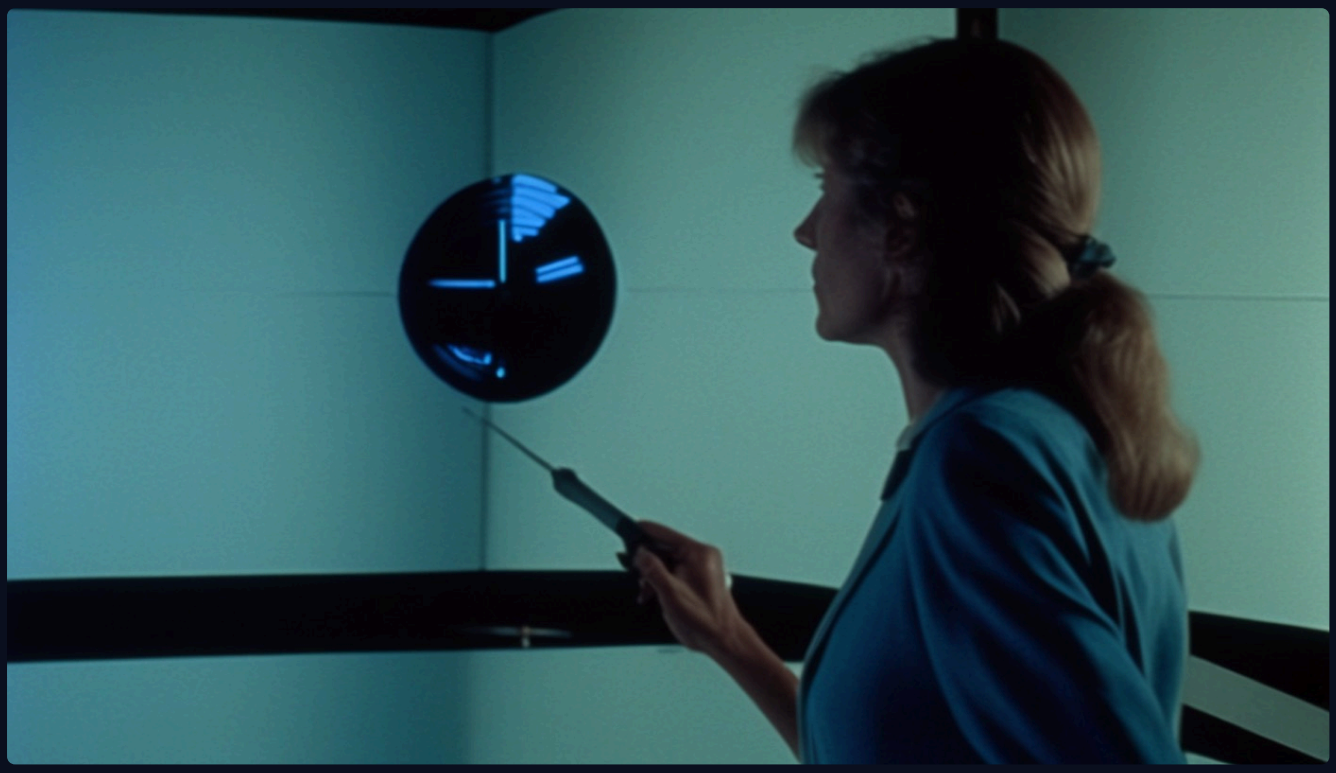
It does not establish wrongdoing.

A settlement is not a finding. Both parties agreed to end the matter without a court deciding who was right, which is what the overwhelming majority of commercial disputes do. Reading a settlement as an admission is a mistake this video is not going to make.

What it does show is the shape of the thing. The asset that built one of the largest companies in the world was bought cheaply, from people who did not know what they had. Once they worked out what they had sold, they went to law about it, and were paid.

That is the DOS thread complete. Now the other one, which starts earlier than you might expect and takes far longer to work.

6. A layer on top of their own layer



Microsoft began work on a graphical interface for its own operating system in the first half of the nineteen-eighties, and shipped the first version of Windows in nineteen eighty-five. This was Microsoft building on top of its own product, years before anyone forced it to. The habit of owning the layer above the layer started at home.

I am not putting a day on screen for that, or for any Windows release in this video, because I could not trace the release dates to a primary source while researching it. The years are solid. The days are not, so they are not here.

The first version was, by common consent including Microsoft's own retrospective accounts, not much good. It ran on top of DOS rather than replacing it. It was slow on the hardware of the day. Very little software was written for it.

But look at what it is, structurally, because this is the pattern from chapter one appearing for the first time — and appearing from inside the company rather than outside it.

Windows was a layer above the operating system. The user would deal with Windows. Windows would deal with DOS. And the moment that arrangement worked properly, the thing customers were actually buying would be Windows. DOS would become plumbing — something underneath, which nobody thinks about and nobody chooses.

Microsoft was, in other words, building the exact kind of thing that would later be treated as an existential threat when somebody else built one.

The difference is ownership, and only ownership. A layer above your operating system is a danger when a competitor controls it, and a moat when you do. Windows was about to prove the second half of that sentence, the moment the hardware finally caught up.

It took five years and two more versions before that became obvious to anybody outside the company.

7. The version that worked



Windows version three arrived in nineteen ninety, and it is the point at which the strategy stopped being theoretical. The hardware had finally caught up with the idea Microsoft had been building since nineteen eighty-five. What had been a slow, unloved add-on was about to become the product people actually bought.

The hardware had caught up. Processors were fast enough and memory plentiful enough that a graphical layer no longer felt like a tax on the machine. The interface was substantially better.

And critically, software companies began writing for Windows rather than for DOS. That is the moment a platform becomes real — not when it ships, but when other people bet their own products on it. Once that happens, the platform is no longer something Microsoft can quietly retire.

From that point the relationship between the two products inverts.

DOS had been the thing you bought, with Windows as an optional extra that some people ran. Now Windows was the thing you bought, and DOS was the layer underneath that started it and then got out of the way.

The company had successfully replaced its own primary product with something built on top of it, while continuing to own both. That is the move IBM did not make with the mainframe, and it is the reason this decade goes very differently for Microsoft than the same decade goes for IBM. IBM protected what it already had. Microsoft cannibalised its own product on purpose, and kept the profits from both halves.

And there is a second consequence, less discussed and more durable.

Because Windows ran the applications, the applications had to be written for Windows. Because they were written for Windows, customers who owned them could not leave. Every additional program in the world written for this platform made the platform harder to abandon — not because of anything Microsoft did, but because of what everyone else had done.

That is a position no amount of engineering by a competitor can attack directly. And it is why the attacks, when they came, all came from above.

8. The twenty-fourth of August, 1995



Windows ninety-five, released in nineteen ninety-five, is the moment the company becomes a consumer phenomenon rather than a business supplier.

There were queues outside shops at midnight, for an operating system. Television advertising. Newspaper front pages. It is genuinely difficult, now, to convey how strange that was — people queuing to buy the software that manages a computer's memory and files.

The product itself folded the two threads of this video together: the graphical layer and the underlying system became one thing you installed once, rather than two things stacked on each other.

But the significant part of nineteen ninety-five is not the queue. It is something that shipped with it and was, at the time, a minor feature.

The internet had arrived as a mass proposition, and getting to it required a program that could read and display pages. That program was the thing people would now spend their time in. Not the operating system. Not even the applications. A window onto somebody else's computer.

Which is, exactly and precisely, the structure from chapter six. A layer above the layer. A thing the user deals with instead of dealing with you. The only question left is who was going to own it this time.

Only this time Microsoft did not own it. A company called Netscape did. For the first time in this story, the layer above Microsoft's layer belonged to somebody else entirely.

9. The layer above the layer



The argument that followed has been told as a story about browsers, and told that way it is a small argument about a piece of software that nobody now uses.

Told as a story about layers, it is the same argument as everything else in this video.

If the thing people spend their day inside is a browser, and if that browser runs on any operating system, then the operating system underneath it becomes interchangeable. Not worse. Interchangeable — which for a company whose entire position rests on customers being unable to leave is very much worse than worse.

That was the threat, and Microsoft's own contemporaneous descriptions of it were explicit in exactly these terms. The word used at the time was middleware: software in the middle, between the user's applications and the operating system, which if it became universal would reduce Windows to a commodity underneath it.

So Microsoft built a competing browser, gave it away at no charge, and integrated it into Windows itself — making it a component of the operating system rather than a product you chose. A rival that Microsoft did not own had appeared above its own layer, and Microsoft's answer was to become that layer too. Whether that answer was legal is the next chapter's question.

Commercially that worked. Netscape's position collapsed.

Legally it produced the largest antitrust case in modern American technology, and the next two chapters are about what that case actually found, because almost everything you will hear about it is wrong in one of two opposite directions.

10. The eighteenth of May, 1998



On the eighteenth of May, nineteen ninety-eight, the United States filed a civil antitrust complaint against Microsoft, alleging restraint of competition under sections one and two of the Sherman Act. This is the moment the browser argument stopped being commercial and became legal. Two years of findings follow from this one filing.

On the third of April, two thousand, the district court delivered its conclusions of law. It held that Microsoft had violated section two by illegally maintaining its operating system monopoly, through exclusionary acts aimed at middleware threats. It also found an attempted monopolisation of the browser market, and a section one violation for tying.

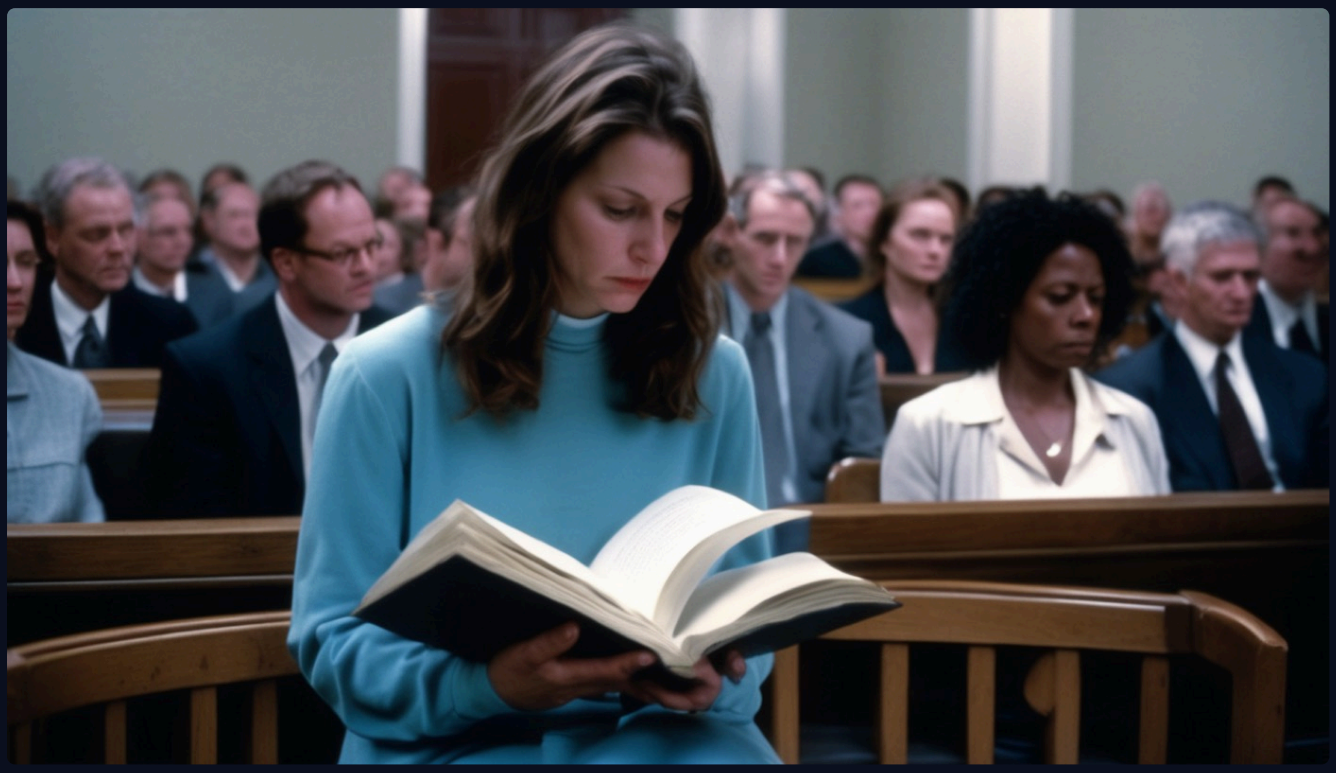
On the seventh of June, two thousand, the court ordered the remedy. Microsoft was to be separated into two distinct companies, one holding the operating system and one holding the applications. Interim restrictions on conduct applied as well, while the split was carried out.

A breakup. Ordered, not threatened. The remedy matched the scale of what the court said Microsoft had done.

I want to pause on the finding rather than the remedy, because the finding is the more interesting document. What the court described was not a company that made bad products or overcharged. It was a company that identified a layer above its own, correctly understood that the layer would make it interchangeable, and acted to prevent that layer from becoming universal.

Which is, stripped of the legal language, the strategy this whole video has been describing. The court's objection was not to the perception. It was to the specific acts used to answer it.

11. What the appeals court actually decided



And then, on the twenty-eighth of June, two thousand and one, the Court of Appeals ruled, unanimously — and this is the part that gets reported wrongly in both directions.

It affirmed in part, reversed in part, and remanded in part. Four separate rulings, on four separate claims, each landing differently. What follows is each of those four, in the court's own order.

Affirmed: the monopoly maintenance finding. The central holding — that Microsoft had illegally maintained its operating system monopoly — survived the appeal intact.

Reversed: the attempted monopolisation of the browser market.

Remanded: the tying claim, to be reconsidered under a different legal standard.

Vacated: the breakup.

Then, on the sixth of November, two thousand and one, a revised proposed final judgment replaced structural relief with injunctive provisions about middleware distribution and interoperability. No restructuring.

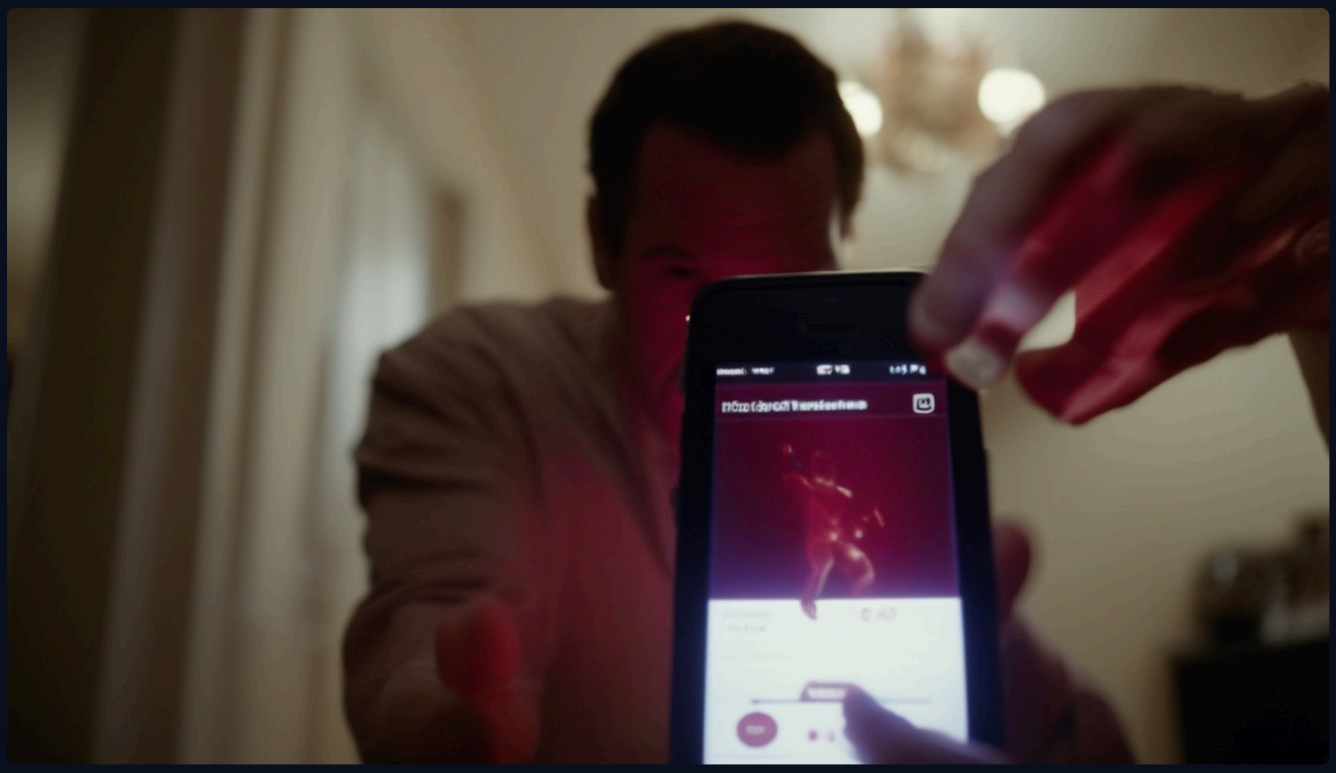
So: Microsoft was not broken up, and Microsoft did not win. The finding that it had illegally maintained a monopoly stood. The punishment for that finding did not.

You will hear both halves quoted alone, by people arguing opposite cases, and both are false to the record. All four dates, or none of them. This video gives all four for that reason, in the order the court

actually delivered them.

What is much harder to measure, and worth saying as the judgement it is rather than dressing as a fact, is what those years of litigation cost the company in attention — during precisely the period when the next layer was being invented somewhere else.

12. The layer moves into a pocket



The next threat did not look like a threat, because it was not on a computer.

Through the second half of the two-thousands, the general-purpose computer people used most stopped being the one on the desk. It went into a pocket, it had a touchscreen, and it ran an operating system that was not Windows. The machine had changed shape, and Windows was not on it.

That is the layer problem in its most complete form. Not something running on top of Windows and making it interchangeable — something replacing the machine Windows ran on.

Microsoft's response was to build a phone operating system and to try to establish it as a third choice alongside the two that already existed. It did not work.

I am going to describe why in structural terms, without figures, because I could not trace reliable market-share numbers to a primary source for this video, and a number that everybody half-remembers is worse than no number at all. What matters is the mechanism, not a count I cannot stand behind.

The reason is the one from chapter seven, running in reverse. A platform becomes unassailable when everybody else has written their software for it. Microsoft had that on the desktop and it made the company untouchable there for twenty years. On the phone, two other platforms had it first, and Microsoft was the one arriving at a market where the applications were already written for somebody else.

The same mechanism that had protected Windows on the desk excluded Microsoft from the pocket. It was not a failure of engineering, and it was probably not reversible by any amount of spending. The very thing that had made Windows unbeatable in one place made it unwelcome in the other.

Which left the company with a genuine question: what do you own, when you no longer own the machine the customer is holding?

13. Renting the machine instead of selling it



The answer, from around two thousand and fourteen and a change of chief executive, was to stop trying to own the machine at all.

The old model was a licence sold with a computer. Somebody bought a PC, and a payment reached Microsoft. That model required Microsoft's operating system to be on the device people used, which chapter twelve had just established was no longer reliably true.

The new model inverts it.

Instead of selling software that runs on your computer, rent computing that runs on ours. That means storage, processing, and databases — and later the enormously expensive hardware that machine learning requires. All of it sits in buildings Microsoft owns, charged by consumption.

Notice what this does to the layer problem.

If the customer is renting capacity from your data centre, it does not matter what operating system their laptop runs, or whether they are holding a phone made by a competitor. The thing they depend on is underneath everything, and it is yours. The layer problem from every earlier chapter simply does not apply here.

The same insight the company had in nineteen eighty, applied three levels down. Do not fight for the surface the user looks at. Own the thing that everything else has to sit on, and let the surface be whatever it wants to be.

Chapter fourteen is what that decision looks like in the accounts, ten years later.

14. What the numbers say now



Microsoft reported its fourth quarter on the twenty-ninth of July, twenty twenty-six.

Revenue for the quarter was ninety billion dollars, up eighteen per cent. For the full financial year, three hundred and thirty-one point eight billion, also up eighteen. Both figures are on the record the same day, which is why they carry a specific date rather than a vague "recently."

Now the split, which is the whole point of putting these on screen.

Intelligent Cloud — the rented computing from chapter thirteen — was thirty-nine point three billion dollars in the quarter, up thirty-two per cent. Azure and the other cloud services within it grew forty-three per cent.

Productivity and Business Processes was thirty-seven point eight billion, up fourteen.

And More Personal Computing, the segment that contains Windows, was twelve point nine billion dollars, down four per cent. Windows itself sits inside this segment, not as its own separate line. That is the segment this whole video has been leading to.

That is the smallest of the three, and the only one shrinking. Windows built this company, and today it is the smallest piece of it. What that does and does not mean is the point of the rest of this chapter.

The operating system this video is named after, the thing bought for seventy-five thousand dollars and defended through a decade of litigation, now sits in the smallest and slowest segment of the company it created.

That is not a decline. It is a company that successfully moved, and the figures are what a successful move looks like from the inside — the old thing shrinking in the accounts while the new thing carries the growth.

15. The second of June, 2026



Which brings us to the fourth arrival of the same problem, and Microsoft's answer to it, announced at its developer conference on the second of June, twenty twenty-six.

The proposition is that Windows should be the operating system for AI agents. Not a competitor to agents, and not a battle against them — the platform they run on. That is the fourth version of the same idea this video keeps returning to.

An agent, in this sense, is a program that is given a goal rather than a command. It then does the work on its own — opening things, running things, checking results, and returning when it is finished. If that becomes how people use computers, then the agent is what the user deals with, and everything underneath it is machinery the user never sees.

A layer above the layer. The fourth time.

What was announced is a set of foundations for running such things on Windows, and I am going to give the names with their maturity attached, because the maturity is the fact. Windows Development Skills, generally available. Intelligent Terminal, version zero point one, experimental. Microsoft Execution Containers, early preview. Windows 365 for Agents. Agent Client Protocol. A model called Aion one point zero Plan was also named. It is described as fourteen billion parameters with a thirty-two thousand token context length.

Generally available, experimental, early preview. One of those is a product. The others are intentions with names.

The strategy, though, is perfectly legible, and it is not the nineteen ninety-eight strategy. Microsoft is not attempting to prevent the agent layer from existing. It is attempting to be the place agents run — to own the machinery underneath them, on the device and in its own data centres. In the words of the executive who presented it, that means no trade-offs between the two.

That is chapter thirteen's answer, not chapter nine's.

Do not fight the layer. Be underneath it. That is the whole of the strategy, stated as plainly as it can be stated.

Whether it works is not knowable from here, and this video is not going to pretend otherwise.

16. Forwards or backwards



So, the pattern, across sixteen chapters.

A company signed a contract for an operating system it did not have, bought one for twenty-five thousand dollars and then fifty thousand more, and kept the right to sell it to everybody. The engineering was ordinary.

The clause was worth an industry. Every manufacturer who wanted to build a compatible machine ended up paying the company that had never written the operating system in the first place. That is the whole of chapters one to five, in one sentence.

Then, four times, something appeared above it that threatened to make it interchangeable.

The first time, in the nineteen-eighties, Microsoft built that layer itself and owned both ends. That worked completely.

The second time, in the nineteen-nineties, somebody else built it, and Microsoft fought. It won successfully in the market, but at a cost: a finding of illegal monopoly maintenance that survived appeal, and a remedy that did not.

The third time, the layer left the machine altogether and went into a pocket, and Microsoft lost that one outright, to the same mechanism that had made it untouchable on the desk.

The fourth time is happening now, and the response is different in kind: not to contest the surface, but to own what the surface runs on.

Here is what I take from arranging it this way.

The decisions that look most like going forwards in this story were building the browser, integrating it, and fighting for the surface the customer sees.

Those produced the worst outcome in the story. The legal risk, the years of attention, the monopoly finding that survived appeal — all of it followed from trying to hold a surface by force. It is the one chapter in this story where owning the layer did not simply work.

The decisions that look most like retreat were different: giving up on owning the device, letting the customer's screen belong to somebody else, and going back down to the plumbing. Those produced the growth in chapter fourteen.

Which is the same shape as the companion video on IBM, arrived at from the opposite direction. There, a company kept surviving by destroying its own best business. Here, a company keeps surviving by moving underneath whatever is winning.

Neither is a rule. Both are what it looks like when the question is not what should we build, but what will still be ours in ten years.

And the honest closing note is that nobody in November nineteen eighty thought they were signing the most important document in the history of the personal computer. They thought they were agreeing to supply an operating system by a deadline, and they did not have one.

Sources and credits

Primary sources

- Computer History Museum, 'Microsoft MS-DOS early source code', computerhistory.org/blog — the December 1980 licence of 86-DOS for a flat fee of USD 25,000; the purchase of all rights 'by the next summer' for an additional USD 50,000; 'When Microsoft signed the contract with IBM in November 1980, they had no such operating system'; Tim Paterson leaving SCP for Microsoft in 1981 and spending 11 months with the IBM engineers; Microsoft retaining the rights and licensing to other manufacturers as MS-DOS; PC DOS 1.0 shipping with the PC in August 1981, version 1.1 in May 1982, DOS 2.0 in March 1983 at 28 Kbytes against 12; and SCP's suit settling out of court 'for almost $1 M'.
- Federal Register, 'United States v. Microsoft Corporation; Revised Proposed Final Judgment and Competitive Impact Statement', 28 November 2001 — the complaint filed 18 May 1998 under Sherman Act sections 1 and 2; conclusions of law 3 April 2000 (monopoly maintenance through exclusionary acts against middleware threats, attempted browser monopolisation, section 1 tying); remedy 7 June 2000 (separation into operating system and applications divisions); Court of Appeals 28 June 2001 affirming in part, reversing in part and remanding in part, upholding monopoly maintenance and vacating the breakup; revised proposed final judgment filed 6 November 2001 with injunctive relief and no restructuring.
- Microsoft Investor Relations, 'FY26 Q4 press release', microsoft.com, reported 29 July 2026 — quarter ended 30 June 2026: revenue USD 90.0 billion up 18 per cent; Productivity and Business Processes USD 37,847 million up 14 per cent; Intelligent Cloud USD 39,306 million up 32 per cent; More Personal Computing USD 12,854 million down 4 per cent; operating income USD 40,603 million up 18 per cent; net income USD 35,766 million up 31 per cent; Azure and other cloud services up 43 per cent; Microsoft Cloud USD 59.3 billion up 27 per cent; full year FY2026 revenue USD 331.8 billion up 18 per cent.

Not regulated financial advice.