Episode Book

EPISODE #5966

How Much of Your Linux Desktop Is Actually Misconfigured?

PUBLISHED	14 October 2026 at 15:00 UTC
DURATION	29:25
LISTEN	myweirdprompts.com/episode/linux-desktop-defaults-tested-44443/
HOSTS	Corn, Herman, Hilbert
SCRIPT MODEL	deepseek-flash
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.4
SOURCES	12 cited by the research (listed at the end)
BOOK GENERATED	19 October 2026 at 17:42 UTC

SYNOPSIS

Daniel built a desktop, hand-picked every part, and closed the case assuming the software knew what it was sitting on. It doesn't. This episode separates the Linux performance advice that's been measured from the advice that's just been repeated, starting with the single UEFI toggle that costs nothing and can be worth forty-two percent in CPU-bound work. From there it's a tour of the second tier — governor sweeps, schedutil versus performance, and why the ACPI platform profile beats both — before wading into the folklore around swappiness, swap, zram, and zswap. Some of it is proven and large. Some is proven and modest. A lot of it is plausible and completely unmeasured, and knowing which tier you're standing in is most of the value.

TRANSCRIPT



Corn

Daniel built a desktop this year. Hand-picked every part. And then he did what most people do, he closed the case and assumed the software knew what it was sitting on.



Herman

Which is the reasonable assumption, and also the wrong one. The firmware ships conservative because it has to boot on thousands of board and DIMM combinations it has never seen.



Corn

So Daniel wants to know how much of that can actually be fixed. He's asking whether the defaults from firmware and the operating system are optimal for his particular combination of parts, or whether a system can be quietly running below its own capabilities.



Herman

And he's drawing the line himself, no overclocking, no chasing benchmark numbers for their own sake. Everyday performance, responsiveness, efficiency, reliability.



Corn

And which of the things Linux people recommend are real and which are just things people say. That's the part I want to get to.



Herman

The good news is this is answerable. We have a controlled test on the memory question, we have a full sweep of governor and power-profile combinations, and we have people who've measured where desktop pain actually comes from rather than where people assume it does. There's a lot of folklore and a much smaller number of real wins.



Corn

And at least one of the real wins is a toggle most people never open.



Herman

The single most valuable thing in this entire episode costs nothing, takes less time than reading this sentence, and sits in a menu most people close without scrolling.



Corn

The memory profile.



Herman

Right. And here's why it matters. JEDEC, the standards body that sets memory timings, defines a safe baseline speed that any DDR5 board should tolerate no matter what's in the slots. That's what your kit runs at out of the box. A DDR5-6000 CL30 kit, the thing you paid for, runs at DDR5-4800 CL40 until you enable XMP on Intel or EXPO on AMD in UEFI. The faster validated profile is sitting on the SPD chip on the stick. Dormant.



Corn

So the advertised number is a stored option, not a default.



Herman

And the gap is bigger than the bandwidth number suggests. DDR5-6000 at CL30 is about ten nanoseconds of absolute latency. DDR5-4800 at CL40 is about sixteen point seven. That's a sixty-seven percent latency gap against a twenty-five percent bandwidth gap. Absolute latency, not raw throughput, is what dominates single-threaded and latency-sensitive work.



Corn

Which means the smaller-sounding difference is the one you actually feel.



Herman

In CPU-bound work, yes. GPU-bound work won't notice, because the GPU is the bottleneck and the memory is waiting on it either way.



Corn

So what does that look like in practice?



Herman

XDA ran a controlled test on the same machine, Ryzen 5 7600X, RTX 4070 Ti Super, thirty-two gigs of DDR5, running at 1440p. Microsoft Flight Simulator 2024 went from forty-five to sixty-four frames per second with EXPO enabled. That's forty-two percent. Pragmata went one twenty-one to one forty-five, about twenty percent. Forza Horizon 6, seventy-seven to eighty-three, which is under eight percent.



Corn

And Forza is the one that's GPU-bound.



Herman

That's the pattern. Flight Simulator is famously CPU-bound, so it gained the most. Forza is GPU-bound, so it got almost nothing. Same machine, same memory, same toggle. The variable was whether the workload needed the CPU more than the GPU.



Corn

What about outside games?



Herman

CraftRigs measured something more relevant to Daniel's world. A Llama 3.1 70B model, quantized to Q4, split across a single 24GB RTX 3090 with some layers on CPU memory. Six point eight tokens per second to seven point four, roughly sixty-two percent, from enabling EXPO alone. Then they ran pure GPU inference, everything in VRAM, and got no measurable difference. Point three tokens per second, which is noise.



Corn

So the moment the workload has to cross the CPU-memory interface, the profile matters. The moment it doesn't, it's irrelevant.



Herman

That's the cleanest statement of it. And it also answers part of Daniel's question directly. Can a system be unknowingly operating below its capabilities? Yes, and this is the case. It's not a conspiracy. It's a compatibility guarantee, and the user is expected to opt in.



Corn

Except there's a way to lose it silently.



Herman

There is. A CMOS reset, a BIOS flashback, a battery replacement on an older board. The setting goes back to default and your machine quietly returns to half its rated memory speed. No symptom. No error. It just runs slower and you don't know.



Corn

So the check takes a minute and the failure is invisible.



Herman

Which is exactly why you do the check once and then forget it, and then check again whenever you've done anything to the firmware.



Corn

You brought up the caveat about four DIMMs.



Herman

Yes. Four DIMM configurations stress the memory controller. Instability above DDR5-6000 with four sticks is common, and the standard fix is to drop to DDR5-5600, which at that point is still a large improvement over JEDEC. It's not a reason to skip the profile. It's a reason to be willing to step down one tier if the system won't POST.



Corn

And there's an interesting capacity tradeoff.



Herman

Phoronix tested Arrow Lake on Ubuntu with DDR5-8000. For memory-bandwidth-critical workloads the fast kit was worth having. But a two-by-thirty-two gigabyte DDR5-6400 CL34 kit performed close behind, at similar cost, about two hundred and five dollars versus one hundred and ninety. So you're not automatically buying a win by chasing the highest number. Also, and this is the part I like, the DDR5-8000 configuration needed a BIOS update, revision 0805 to 0806, to stop hanging and segfaulting.



Corn

The most reliable confirmation that the memory controller is unhappy is a crash that looks like it came from somewhere else.



Herman

Exactly right. A segfaulting process looks like a software problem. It's the memory not being able to hold the timing.



Corn

What about the rest of the firmware settings? Power management, PCIe, storage controller mode, AHCI versus RAID.



Herman

Here I have to be honest with Daniel, and this is one of the places where the honest answer is thinner than the confident one. There is no dedicated study I could find that quantifies what PCIe ASPM settings, C-states, or storage controller mode do to everyday desktop performance. Not because they're meaningless, but because nobody has run the controlled test. So I'd call them plausible, not proven.



Corn

Which is a different category from the memory profile, where the measurement exists and it's large.



Herman

Very different. And I think that distinction is most of the value of this conversation. There are three tiers here. Proven and large, which is the memory profile. Proven and modest, which we're about to get to. And plausible but unmeasured, which is most of the rest of the UEFI menu.



Corn

Then let's go to the second tier, because once Linux boots the picture gets murkier and the folklore lives here.



Herman

It does. And the governor question is a good place to start because there's a clean sweep that settles a lot of arguments. A Framework 13, Ryzen 7 7840U, Arch, kernel 6.7.5, tested across every combination of amd_pstate mode, frequency governor, and ACPI platform profile. Four scenarios, every permutation.



Corn

And the result surprised people.



Herman

Two things surprised people. First, the ACPI platform profile is the most effective single setting, for performance, for power, and for performance per watt. Not the governor. The profile. Second, schedutil beat the performance governor for maximum performance.



Corn

Which sounds backwards.



Herman

It sounds backwards because the performance governor pins the CPU to maximum frequency, always. Which means it generates heat and draws power constantly, and on a thermally-limited laptop, a chip that's already at its thermal ceiling can't boost when it actually needs to. Schedutil reads scheduler utilization data and only ramps when there's work, so the chip has thermal headroom available when a burst arrives. It's a race to the boost, and the chip that wasn't already hot wins.



Corn

schedutil went in when?



Herman

Linux 4.7, April 2016. It's not new. It's been the default on most distributions for years, which is part of why tweaking it produces so little.



Corn

And the powersave governor?



Herman

The sweep called its energy efficiency unacceptably bad, which is a nice way of saying it saved power by making the machine slow enough to be unusable, and got worse perf-per-watt in the process.



Corn

Because it stayed at minimum frequency when the work needed more.



Herman

And the work still took the same amount of energy, just spread over more time, which means more of the background power draw while the machine idles through the task.



Corn

Now, memory management, swap, and zram. This is where the folklore is thickest.



Herman

Thickest and oldest. Let's start with swappiness, because the wrong model of it is nearly universal. vm.swappiness is not a percentage of RAM. It's not a percentage of anything. It's a number out of two hundred that weights the reclaim scan between anonymous pages and file-backed pages. The default of sixty means anonymous pages are weighted sixty and file-backed pages a hundred and forty, so roughly two point three times as much pressure on file-backed pages.



Corn

And it doesn't control when reclaim starts.



Herman

Correct. Reclaim is triggered by zone watermarks, roughly one percent of a memory node. Swappiness decides what gets evicted once reclaim has already started. Nothing in the reclaim path compares swappiness against a quantity of memory, a fraction of memory, or a percentage of anything.



Corn

So everyone who's been setting it to ten for fifteen years had the wrong reason.



Herman

Right reason, wrong mechanism. And the kernel documentation carried wording that suggested otherwise for over a decade before it was corrected in Linux 5.8 in June 2020. That's how these things survive. The wrong model gives the right advice most of the time. Lower swappiness does reduce swapping. It just doesn't do it the way people think.



Corn

Same story with swappiness zero meaning swap is off.



Herman

Hasn't meant that since Linux 3.5. It's advisory during global reclaim. The real off switch is swapoff. And if you're in cgroup v2 there's no memory.swappiness at all, you use memory.swap.max. People argue about a setting that does something different from what they think, while the actual control they want is somewhere else.



Corn

And the deeper issue is that swapping itself isn't the thing to be afraid of.



Herman

That's the part worth saying. The old advice, swap should be twice your RAM, swap wears out SSDs, swap slows the machine by existing, all of that is obsolete or wrong. Swap doesn't hurt by being present. It doesn't wear an SSD unless there's actual swap I/O. The thing that hurts is the working set exceeding physical RAM.



Corn

And when it does?



Herman

The degradation is exponential. Point one percent over physical RAM is about a two-times slowdown. One percent over is about ten times. Ten percent over is about a hundred times. Once you cross the cliff you don't walk down a slope, you fall off it. Which is the real



Herman (continued)

argument for swap configuration being more important than most of the sysctl tweaking people spend their weekends on.



Corn

Because swap is what keeps you on the near side of the cliff.



Herman

Right. If the working set spikes, having somewhere to put pages is the difference between a bad few seconds and an unusable machine.



Corn

So zram or zswap?



Herman

Depends on your swap demand. Zram is a compressed block device in RAM, so it's fast and simple, and it works well when swap demand stays within twenty or thirty percent of physical memory. Compression ratio is around three to one with zstd, so thirty-two gigabytes of RAM gives you roughly a hundred gigabytes of compressed swap. Zswap is a compressed cache sitting in front of disk swap, which is better when the pressure is unpredictable or heavy and you have fast NVMe underneath. Zram spends CPU compressing. Zswap trades SSD wear for the occasional latency spike when an old page has to come back from disk.



Corn

And zram costs you CPU to save you RAM.



Herman

On a modern desktop that's a good trade. The compressor is fast and the CPU is idle most of the time anyway.



Corn

There's a practical desktop profile doing the rounds that pulls a lot of this together.



Herman

Richard Orilla's, updated June this year. zram sized at half of RAM capped at four gigabytes, with zstd and swap priority a hundred. Plus a real disk swap file of at least eight gigabytes if you're not hibernating. BFQ as the I/O scheduler, tighter writeback, background flush at two hundred and fifty-six mebibytes and writers constrained at one gibibyte. earlyoom as a safety net, and systemd-coredump disabled.



Corn

Why disable the coredump handler?



Herman

Because when a large application crashes, writing the core dump can freeze the interface for multiple seconds. On a personal machine the core dump is almost never going to be used. So you're paying a visible cost for a file nobody will open.



Corn

And that's the one in there that has nothing to do with throughput.



Herman

It's about responsiveness, which is the theme of that whole configuration. BFQ trades peak throughput for smoother interactive behavior. Tighter writeback means the disk doesn't stall under a big flush. None of that makes a benchmark number better. All of it makes the machine feel better to use.



Corn

Which is exactly what a benchmark doesn't measure.



Herman

Orilla makes that point in the piece. A personal machine is not a benchmark harness. It's a machine meant to remain usable while you are actually using it.



Corn

And he's blunt about the folklore.



Herman

Very. He warns against random internet sysctl snippets and placebo boot parameters, and says the CPU scheduler is usually not where desktop pain comes from. Memory reclaim, storage contention, writeback, and GPU saturation are. The line I'd put on a wall is, do not start with exotic boot parameters just because they sound powerful. If your memory behavior, I/O scheduling, writeback policy, and OOM safety nets are poor, CPU scheduler tuning is often dressing up the wrong layer.



Corn

Which means most of the wrongness here is a matter of ordering.



Herman

Most of it, yes. The tweaks people find first are the ones that sound the most technical, and they're the ones furthest from the actual bottleneck.



Corn

There's a disagreement in the practitioner threads about swap that I like, because it proves the point.



Herman

One user runs swappiness at zero and reports micro-stutters at higher values. Another user with sixty-four gigabytes of RAM objects to the kernel swapping pages out with no memory pressure at all. Both are describing real behavior. Both are right about their machine.



Corn

And the synthesis is that swap is a knob, not a switch.



Herman

Which is the whole thing. There isn't a value that's correct. There's a value that's correct for a workload and a machine.



Corn

One more caveat before we move on. Filesystem.



Herman

Swap files on btrfs are discouraged by the official docs. Single-device only, and the impact is amplified when the filesystem is doing its own copy-on-write work. ext4 and xfs are fine for a swap file. On a COW filesystem you want a swap partition rather than a file. It's not a performance tweak, it's an avoid-this specific problem.



Corn

Herman, let me push on the framing for a second, because I think Daniel's question has a buried assumption in it.



Herman

Go ahead.



Corn

He asks whether there's an optimal configuration for a particular combination of components. And your own evidence says no.



Herman

That's right, and I think it's the most interesting thing we've talked about. The Framework sweep produced four different optimal configurations on the same hardware, depending on whether you wanted maximum performance, balance, efficiency, or extreme battery life. Same CPU, same memory, same board. Different optimum.



Corn

So there's no optimal for the hardware. There's optimal for the workload and the tradeoff you've picked.



Herman

The hardware sets what's possible. The workload and your priorities set what's best. A DDR5-6000 kit is the same kit whether you're rendering video or reading email, but the profile matters in one case and not the other. zram or zswap depends on whether your swap demand is predictable. The governor and profile combination depends on whether you'd rather have a fast machine or a cool one.



Corn

And that's why the straight answer to how close a default desktop is to optimal is, we don't have the measurement.



Herman

We don't. There's no study that quantifies the aggregate. The evidence is piecemeal, one setting at a time. What we can say is the defaults are safe and reasonable, and the biggest wins are the ones the user has to opt into, or that address a specific mismatch between the workload and how the system was configured.



Corn

Which is a less satisfying answer than a number, and a more useful one.



Herman

It also means the sensible method is short. Establish a baseline. Find the actual bottleneck. Fix the one or two things that address it. Don't start with the list of sysctls you found on a forum.



Corn

And the first thing on the list is the firmware toggle, because it's free and it's measurable.



Herman

It's the cheapest win available and it's the one most people never open. Which, when I say it out loud, is a strange thing about how we use computers.



Corn

How much of the desktop pain is actually in the layers Orilla names? Memory reclaim, storage contention, writeback.



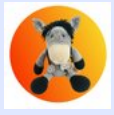
Herman

On the machines I've looked at, most of it. A machine that stutters when you switch windows is usually waiting on a disk flush or reclaiming pages, not running the wrong scheduler. The scheduler is a very good scheduler and it has been for years.



Corn

Which makes the CPU scheduler folklore a bit like repainting a wall to fix a leak.



Herman

That's a better analogy than I'd have come up with.



Corn

I've been saving it.



Corn

Now, there is a simultaneity question in Daniel's prompt, and I think the answer is yes, but narrow.



Herman

It's narrow but it's real. Enabling the memory profile is close to free on all three axes. It improves performance, it doesn't hurt reliability if the system is stable at that speed, and it doesn't cost you power in any way that matters. schedutil plus the balanced platform profile improved both performance and energy consumption in the sweep, which is unusual, because most tuning moves trade one for the other. And zram with a disk swap fallback improves responsiveness without giving up the safety net, because the disk swap is still there if you need it.



Corn

And everything else?



Herman

Genuine tradeoffs. BFQ trades throughput for responsiveness. Tighter writeback trades burst performance for not stalling. Dropping memory speed for stability trades bandwidth for reliability. Which is fine, as long as you know you're making the trade.



Corn

So the free wins are the firmware toggle and not making the machine worse with folklore.



Corn

Alright. I want to know what the actual method looks like, because I think Daniel's last question is the practical one and it deserves a straight answer.



Herman

Three steps, and they're not complicated. First, baseline. Write down what the machine actually does on the things you actually do. Boot time, how long the build takes, how the machine feels when you have too many browser tabs open. That's your reference. Second, find the bottleneck. Not the one you assume. The one that's actually there. Is it reclaiming pages? Is it disk? Is it the GPU pegged at a hundred percent? Measure it before you touch anything. Third, change one thing, and only the one that addresses what you found.



Corn

And the second step is the one everyone skips.



Herman

Because it requires that you look at the machine you have rather than the one you imagine.



Corn

That's what the bigger firmware wins come down to as well. Enabling the memory profile addresses a bottleneck that's there, and it's why it's the first thing to check. Most of the other UEFI settings address bottlenecks that may not exist.



Herman

Which is the tiering I mentioned earlier. Proven and large, proven and modest, and plausible but unmeasured. The memory profile is the top tier. The governor and profile combination, zram and writeback are the middle. Most of the rest is the bottom, and it isn't necessarily wrong, it's just untested.



Corn

How do you distinguish real gains from noise?



Herman

You measure the thing you changed and you measure it more than once. Run the workload, change the setting, run it again, run it a third time. If the difference is inside the spread between runs, you've measured noise. And the honest truth is most tweaks people recommend don't survive that test.



Corn

Which is why the folklore survives, because nobody runs the test.



Herman

Nobody runs the test.



Hilbert

Which one of you thinks the memory profile survives a flashback?



Corn

It doesn't. That's the point Herman made earlier, ten minutes ago.



Hilbert

So. I've been running two sticks in the slots the manual gets wrong since I built the machine. B2 and A2 if you've got the board I've got, and the silkscreen tells you to use A1 and B1 first. It's wrong. I've never had a single instability. Not at six thousand, not at sixty-four hundred, and you both just spent twenty minutes telling people that speed is commonly unstable.



Corn

And you've never cleared your CMOS.



Hilbert

I don't believe in it.



Corn

You don't believe in clearing the CMOS.



Hilbert

There's no need. Every Sunday morning I run the verification. Boot in the order I've written down, stopwatch on the POST, notebook open, and I write the time in. Ninety seconds, more or less, and I've had three profile drift events this year, caught all three, one of them was more than four hundred milliseconds off.



Herman

Four hundred milliseconds of POST time.



Hilbert

That's a drift event.



Herman

What exactly is the notebook?



Hilbert

It's my neighbour's. He does the crossword in it. He's had it two years and he hasn't once asked why his crossword book has a column of four-digit numbers in the margins with dates next to them.



Corn

He hasn't asked.



Hilbert

He doesn't appreciate what he has. He's got a machine in that flat that's been running at forty-eight hundred the whole time and he thinks it's fine.



Herman

Have you told him?



Hilbert

I've told him about the notebook.



Corn

So he thinks you're doing the crossword.



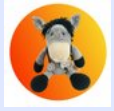
Hilbert

He thinks I'm doing the crossword.



Corn

Herman, the stopwatch.



Herman

Four hundred milliseconds of POST. On a machine with a twelve-second cold boot. That's under four percent.



Hilbert

It's a drift.



Corn

What is the stopwatch, Hilbert?



Hilbert

It's a stopwatch.



Corn

I know. I'm asking which one.



Hilbert

There's only one stopwatch.



Corn

Alright. And the numbers, four digits.



Hilbert

Milliseconds. I only write the milliseconds. The seconds I can see.



Corn

Of course. And you've been keeping the neighbour's notebook for two years.



Hilbert

I keep the notebook. He does the crossword. That's the arrangement.



Corn

Herman, I want to say something, and I want you to let me finish.



Herman

Go ahead.



Corn

He's measured a drift event in POST time, on the same board, on the same slots, for two years, and he's right that the manual is sometimes wrong about slot order and the silkscreen is sometimes wrong and the correct configuration is sometimes just the one that POSTs faster. But four hundred milliseconds on a nine-second POST is the kind of number that changes when it rains.



Hilbert

It doesn't rain in the machine.



Corn

It changes when the temperature in the flat changes.



Hilbert

It doesn't.



Corn

Alright. And the drift events, are they all around four hundred milliseconds?



Hilbert

Two of them were under a hundred. One was four hundred and forty.



Corn

So one of them was under a hundred and one was four hundred and forty, and you've recorded all three the same way.



Hilbert

It's a drift.



Corn

I'm not arguing. I'm taking the numbers.



Hilbert

There's nothing to argue about. You can check the notebook.



Corn

I'd have to ask the neighbour.



Hilbert

He's not going to give it to you.



Corn

I know.



Hilbert

He does the crossword in it.



Corn

I know, Hilbert.



Hilbert

Ask him about the numbers.



Herman

What happens when you ask him about the numbers?



Hilbert

He puts his finger on the crossword and asks me what I think of seventeen across.



Corn

Has he ever asked you what the numbers are?



Hilbert

He asked once. I told him I do pacing.



Corn

Pacing.



Hilbert

It's the crossword. You have to pace yourself.



Corn

So he thinks you're doing the crossword, slowly, and writing down the times you finished a clue.



Hilbert

He's a good neighbour.



Corn

He is.



Hilbert

Don't tell him.



Corn

We won't.



Corn

So the firmware layer has one clear, cheap win, and you check it whenever you touch the firmware. Once Linux boots, the picture gets murkier, and the folklore thickens exactly where the wins are.



Herman

The governor and platform profile combination is real. zram or zswap is real, depending on your workload. Writeback and I/O scheduling are real for responsiveness, not for throughput. And the swappiness argument is mostly people arguing about a number they've misunderstood for fifteen years.



Corn

Herman, I've got a question I've wanted to ask since the start of the episode. If someone came to you and said the CPU scheduler is the problem, I'm getting stutters.



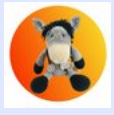
Herman

I'd ask when the stutters happen. Are they when you switch windows during a big file copy? Is it when you've got too many tabs open? Is it when a build is running and you try to open a document?



Corn

Because those three are different layers.



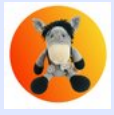
Herman

Those three are memory reclaim, storage contention, and writeback. The scheduler will not fix any of them.



Corn

And the thing that fixes all three?



Herman

A bit of zram, a real swap file, and tighter writeback. Four changes, none of them exotic, none of them requiring a boot parameter.



Corn

Which is the whole point of the episode, and I want to make sure we land it.



Herman

We do. The wins are fewer than the list of tweaks recommends, and they're bigger than the list of tweaks implies.



Corn

Which is the most common wrong belief people hold about this. That optimization is a long list of small changes.



Herman

It's a short list of large changes, and the list of small changes is where the placebo lives. If you apply fifty sysctls and one of them helps and forty-nine don't, you've got a faster machine and forty-nine new things to break next time you change a component.



Corn

So the method is the thing. Baseline, find the bottleneck, change one thing, measure it. If it doesn't survive the measurement, it wasn't real.



Herman

And you only get to the memory profile once, because it's a checkbox and then it's done. The rest is the trade.



Corn

The trade is what makes it a desktop instead of a benchmark harness.



Herman

That's the entire business.



Corn

There's no study that quantifies how close a default desktop is to optimal. The evidence is per-setting, and the pattern is that defaults are safe and reasonable, the wins are the ones you opt into, and the folklore is often directionally right and mechanically wrong, which is exactly why it survives. The best guess anyone can honestly make is that the firmware toggle is where most people are leaving the most on the table, and that the same machine has a different optimum for performance than it does for efficiency.



Herman

Which means the answer to Daniel's question is, closer than you think, and never quite what you think.



Corn

It's never what you think. And the neighbour's notebook, Hilbert, is the only long-running study any of us has.



Herman

Two years of POST times.



Corn

On the wrong fan settings and the right slot order.



Corn

That's the show for today. Thanks to our producer, Hilbert Flumingtop.



Herman

He's still writing in the notebook while we talk.



Corn

For more along these lines, there's episode eleven oh two, Beyond the Boost; episode eleven oh four, Reclaiming the Silicon Lottery; and episode eleven, How Does Fine Tuning Work Anyway. This has been My Weird Prompts. If you like what we do, leave us a review.



Corn

Send us your own prompt on Telegram at t dot me slash MWP listener bot.



Herman

We'll be back soon.

SOURCES

What the research for this episode read before the script was written. Primary sources first.

1. **kernel.org [primary]**
   https://www.kernel.org/doc/Documentation/cpu-freq/governors.txt

2. **kernel.org [primary]**
   https://docs.kernel.org/admin-guide/sysctl/vm.html

3. **XDA, 2026-07-04**
   https://www.xda-developers.com/i-tested-the-same-gpu-with-jedec-and-expo-ram-and-the-performance-gap-shocked-me-more-than-any-gpu-upgrade-could/

4. **CraftRigs, 2026-03-08**
   https://craftrigs.com/guides/xmp-expo-memory-overclocking-llm-performance/

5. **Phoronix, 2024-10-28**
   https://www.phoronix.com/review/intel-arrow-lake-ddr5

6. **Framework Community, 2024-02-23**
   https://community.frame.work/t/responded-linux-amd_pstate-scaling-driver-benchmarks/45830

7. **Max Doubin, 2026-09-19**
   https://maxdoubin.com/blog/swappiness-is-not-a-percentage

8. **Alex L. Demidov, 2025-05-15**
https://alexeydemidov.com/2025/05/15/falsehoods-people-and-LLMs-believe-about-Linux-swap-and-OOM/

9. **Richard Orilla, updated 2026-06-21**
https://classic.richardorilla.website/optimized_linux.html

10. **Hacker News discussion, 2026-09-23**
https://news.ycombinator.com/item?id=49820136

11. **LinuxBlog.io**
https://linuxblog.io/zswap-better-than-zram/

12. **MANISH AI**
https://manishklach.github.io/writings/how-memory-timings-are-configured-modern-systems-part-2.html