

[← Back to Kevin's newsletters](#)

Published: 2020 Aug 8

Emigration and other hedges

When my partner and I moved to Taiwan in March, many of our friends were quite surprised. Back then, all I wrote was: “**long pandemic story short, ended up in Taiwan**”.

Today I want to share that story in more detail, for two reasons:

1. To encourage you to *make decisions*, even (especially!) when you are uncertain about low-probability extreme outcomes.
2. To remind you that emigration *maybe* an option to hedge against such outcomes.

I emphatically do *not* mean to say that you should emigrate somewhere, nor imply that everyone even has this choice. (Of course not! Just like the goofy computing shit we usually discuss on this newsletter, emigration is not available to literally billions of people due to systemic political, social, and economic factors entirely outside of their control.)

I'm writing to you, my friend, so that if you are fortunate enough to have the option, you at least consider it.

A digression on earthquakes

Before we get to emigration, let's first discuss reasoning about low-probability extreme outcomes.

My advice, in short:

1. Resist the urge to round small numbers down to zero,
2. because a small number times a big number might still be very big.

Here's an example: One day in my 20's I realized I was living on the **Cascadia subduction zone**, so I decided to park two weeks of non-perishable food, water, and basic medical supplies in my backyard.

Did I think an earthquake was imminent? No.

Did I think that after an unlikely (but possible!) earthquake, I might be on my own for a while, and so might appreciate past-Kevin's bottled water and beans foresight? Well, yeah, I saw **Katrina** on television, and there's no evidence to suggest that wouldn't happen to me. So, spend 0.1% of my annual income once every decade (gotta replace the beans) to hedge that risk? Absolutely! Yes! Good trade!

This isn't paranoia or disaster fetishism. I'm an optimistic, happy person; I don't spend my evenings lovingly staring at a horde of gold bars and shotguns.

However, I *do* appreciate a good back-of-the-envelope calculation and once heard that **Black Swan Guy** on a podcast. So I'm aware that "unlikely" is not the same as "impossible" – and the distinction matters when the stakes are high!

(Homework: If you live in an area at risk of earthquakes – e.g., anywhere along the American West coast – do you have two weeks of food and water? If not, please stop reading now and get some. Seriously, 2020 is barely halfway over. While you're up, practice closing your house/apartment's gas line too. I mean it! I don't want to hear from any of you ding dongs about "local-first software" or "distributed systems" until you have a local-first supply of water and carbohydrates.)

So, how did I end up in Taiwan?

To explain, let me set the scene; come back with me to the *before times*.

Early February 2020, I meet with a friend visiting town. He asks if I'm worried about this virus in Asia. "No, I don't really spend time following the news." He says no more about it. We eat burritos.

Late February, I'm in New York City visiting friends on my way to a small conference. I have beers with my **Instagram-famous uncle** and he wonders if this virus thing might affect his travel plans. A woman in a fancy soap store asks if "in the spirit of the times" we'd like to wash our hands with her wares. My partner and I travel on to our conference.

The first week of March, we conference. There is much discussion of technical matters, old friends, drinking, new friends, and I continue to pay no attention to the Coronavirus.

Second week of March, I'm working onsite with a client in the finance industry. On Monday March 9, we watch the S&P 500 trip its circuit breaker for falling too quickly immediately after trading opened. Now I (and seemingly everyone else in the West) start paying attention.

By Wednesday, the severity of the situation is made clear when one of the few remaining American institutions trusted by the general public acts against its short-term economic interest: The NBA **stops playing basketball**.

I'm telling you all this to set the context: I'm in the middle of a one-week client engagement on a small British Overseas Territory and three days into paying any attention whatsoever to Coronavirus. Like everyone at the time, I have no idea how the virus will spread, its health impacts on individuals, the social and economic impacts, etc., etc.

Fortunately, our travel plans — a 2-day layover / vacation in Panama on our return to the US — now seem like an obviously a bad idea, so my partner and I realize we need to make alternative arrangements.

Here's where reasoning about low-probability extreme events comes in. I, a non-expert, non-disaster-enthusiast, can imagine only two extreme scenarios to worry about:

1. getting sick ('rona or otherwise) and being unable to access medical care
2. supply chain breakdown (i.e., empty grocery stores) and likely social consequences (violent theft, etc.)

Again, I didn't think these were *likely*, but I didn't think they were *impossible* either. I guesstimated the odds of Safeway closing for two weeks at 5%.

Given these two broad low-probability, high-badness concerns, that Wednesday my partner and I discussed a few possible options:

1. Stay put. This was tempting: My client very generously offered to host us and, as you might expect from someone who works in finance from a British Overseas Territory, they have a very comfortable home. However, if international travel shut down we'd be stuck on a small island. Tourism was obviously over for at least a few months, which was not going to be great for the local economy. If supply chains got cut off, we wouldn't starve (lots of fishing boats), but the rum and diesel would run out quickly. As for the pandemic itself, well, the Four Seasons medical clinic I'm sure has great food but, uh...
2. Head back immediately to Oakland (where my partner lived at the time). I wasn't thrilled about this option, as the San Francisco Bay Area *normally* feels like it's on brink of collapse. I've witnessed (in broad daylight, in multiple places!) people publicly defecating (in front of police!), breaking into cars, and shooting up. Yes, the reasons for this are complex and tragic, but that longstanding dysfunction suggested to me the Bay Area might not be the best place to be during a pandemic. (The anti-vaccine literature prominently displayed in every wealthy Berkeley grocery store checkout queue didn't instill confidence either.)
3. Head elsewhere in the US. We briefly considered this, but had no evidence to suggest that any particular family member, friend, local government or region would fare better than another. Besides, heading anywhere else would put us at a disadvantage in terms of the supply chain risk (my partner's beans were in Oakland).

More broadly, the US national response at that time was, uh, uninspiring (basketball association excepted), and I was concerned about how the population as a whole might react. (I suspect more Americans have guns than earthquake kits.)

On Thursday, March 12 it occurred to me that the best place to go might be in Asia, for two reasons:

1. Many Asian countries already have a social norm around wearing masks in public (partly from the first **SARS** outbreak and partly because, well, they consider it polite not to sneeze on people).

2. Asian countries were “ahead” of the West in terms of the pandemic, so instead speculating as to how a Western government *would* react it was possible to see how Asian governments *were* reacting.

Okay, sure, but which Asian country should we actually head to?

I figured it'd be better to be in a wealthier country, which ruled out places like Vietnam, Thailand, etc.

Hong Kong seemed to be handling the virus well, but also **losing its autonomy**. Singapore is famously well run and also famously **not so free**. Japan I'd visited (and quite liked!), but they **seemed to be in denial** about the virus due to their hosting of 2020 Summer Olympics. South Korea I'd always wanted to visit and was already testing extensively (via **drive-throughs!**).

However, in my hour of Googling I found this **this JAMA article** which made it clear that Taiwan would be our best bet due to the *extremely* competent government response thus far. (Which I experienced firsthand when I **arrived in March**.) For example, Taiwan began screening flights from Wuhan in *December* and in February the **military was deployed to work in mask factories**. Also: The *vice president* (at the time, now recently returned to academia) was the epidemiologist who oversaw the health department during the 2003–2004 SARS outbreak.

Again, not an expert, but, uh, democratic island nation with strong ties to the US, national health insurance system, extensive domestic medical manufacturing capacity, and an *epidemiologist vice president*? Yes, good place to be for a global pandemic!

On Friday March 13 we bought tickets: \$700 per person from Puerto Rico to Taipei (including a no-change-fee “please god just book a flight, our industry is melting down” return leg to San Francisco).

Once we realized we needed to make a decision, it was actually pretty easy because of the payoff matrix skew. The potential downsides were, what? We get rejected at the border and end up having wasted a flight? The virus turns out to be no big deal and we feel dumb for having spent 5% of our income on an unnecessary working-vacation?

The potential upside? Well, let's just say that while I've spent the last 4 months wandering freely around Taipei eating¹ **gua bao** with my 23 million new friends, I've also been watching all my old friends growing increasingly bad haircuts at home while trying to avoid permanent organ damage.

Why am I telling you all this?

I'm not sharing my story because I want to gloat. I'm sharing my story because *all the reasoning still holds*.

Like, there is still a pandemic! You are more likely to get COVID *now* than you were back in March when everyone was flipping out. You won't make the news anymore, but it will still be very bad for you and your family!

On the other side of the equation, how difficult is it to move to another country?

Clearly, that depends on your situation: Can you work remotely or afford to be out of work for a while? Do you have the literal privileges granted from the governments involved? Etc.

In our case, we entered Taiwan as visa-exempt American citizens. We didn't plan to move permanently (we had only our conference trip duffel bags), but after a week of watching the US response it felt likely we'd want to stay our full 90 days² in Taiwan.

My partner offered her landlord upfront payment in exchange for reduced rent, but after that was declined she ended her lease and hired a freshly unemployed family-member to move everything into storage. (Aside from the cost-savings of not paying for an unoccupied apartment, we also figured the Bay Area rental market would be cheaper a few months into a pandemic anyway.)

After we'd finished with our two weeks of arrival "self-health isolation" and spent a few weeks moving freely in Taipei, we decided to explore our options for staying longer. The cost of this hedge also turned out to be quite low for us: We were issued 3-year residency permits after

submitting paperwork online showing we'd been paid salaries of at least US\$5,300/month sometime in the past three years.

(Note: This “**Gold Card**” program is still active and one can apply from outside of Taiwan. The government also considers **studying Mandarin** to be a valid reason to enter Taiwan.)

I feel more comfortable describing my situation as a “hedge” than as “emigration”. The latter word somehow feels more permanent; more serious. My mother, for example, as a teenager jumped on a fishing boat the morning after her city fell to **invading communists**, promptly got lost in the ocean, and was eventually rescued to America.

I, on the other hand, watched three movies back-to-back and took a nap.

Hedging is all about balancing hard-to-know risks with (sometimes) hard-to-know costs. In terms of risks, we've mostly discussed my experience with the pandemic, but there are of course others.

E.g., my fellow Americans, I'm sure you've noticed our police³ **erupting in violence**. They're pulling people into unmarked vans **coast to coast**. Our president has been calling my hometown a **beehive of terrorists** and speculating about **delaying the election**. It's not exactly a “healthy democracy” look!

Now, I'm not saying you should or shouldn't emigrate, quarantine, protest, or whatever. I just want to remind you to actually *think* about low-probability events and extreme outcomes.

What you decide to do is, of course, quite personal. How can you best take care of yourself and your loved ones? How should you balance your idealism with your pragmatism? Is it better to try to reform a government/society or contribute to one that already better reflects your values? **Voice or Exit?**

Hard questions, yes, but worth thinking about!

And if I, your newsletter nerd friend, can help in any way, please reach out. Please stay safe, do good, and be well.

Okay, now back to dumb computer stuff.

A very thin keyboard

Behold, a very thin keyboard!



The joy of this project is inventing and balancing contradictory goals. Let's make a keyboard that's both *mechanical* (with keys that have a satisfying amount of travel) but also *thin* (so not too much travel). It should be usable (easy to learn, type fast, be reliable) but also small and portable (wireless, fewer keys).

This is product design as *sport*. The keyboard doesn't matter, just as picking up a heavy steel bar or reaching the top of a mountain doesn't matter. What matters is the struggle!

Luckily, in aiming to make the thinnest possible mechanical keyboard (no thicker than the switches themselves), I've ensured myself a sufficient supply of struggle. I've had to source custom thin lithium ion cells, design protection circuitry, and select the rest of the board components to minimize leakage currents and other standby losses. On the firmware side, I've had to dive into the details of bluetooth and evaluate architectural tradeoffs between latency, battery life, and ease of use.

So far, my work has been largely theoretical – rough back of the envelope calculations based on datasheets and YouTube videos. But I recently purchased **an oscilloscope** (an instrument for measuring voltages on the timescale of microseconds), which will let me test various assumptions about my code and hardware. E.g., exactly how many cycles does it take to process a keypress and go back to sleep?

Rough firmware has also allowed me to test actually *using* the keyboard. (Yes, it works!) After just a few minutes, several industrial design issues became apparent.

The first you probably already noticed: The USB charging cable's rubber strain relief lifts keyboard off the table. I'd anticipated this when I

specified the mid-mount connector, and decided I'd figure it out later. Well, later is now! I can think of a few possible solutions:

- Ignore it; after all, the cable is only needed for charging, so if the battery lasts sufficiently long it's not a big deal.
 - Design an enclosure so that the circuit board is lifted up the extra millimeter or two. Ugh, not sporting!
 - Shave down the cable thickness. Practical!
 - Drop the USB connector entirely and use wireless charging.
- Hmmmm...

It's really small too!



The other issue that became immediately apparent once I started using the keyboard is, uh, the number of keys: 5x5. This is many fewer keys than a normal keyboard.

Again, the constraint here was chosen primarily for sport (also because, if I recall correctly, it cost an extra \$10 for PCBs larger than 100mm square).

The design tradeoff here is between usability and size.

Perhaps the easiest keyboard to learn would have a one-to-one mapping between keys and characters, but such a keyboard would be quite large. At the opposite extreme, perhaps the smallest, most portable keyboard would be a wireless telegraph key, which would both require a lot of training (learn Morse code!) but also have a low typing speed ceiling.

Chorded keyboards like **stenotypes** have fewer keys than a normal computer keyboard and rely on operators to press multiple keys simultaneously. The current world typing record (**360 words per minute**) is on such a machine.

Traditional keyboards reduce number of physical keys needed by moving symbols and uppercase letters to a separate “layer” that’s accessed via the “shift” key. But there’s nothing special about one layer; in the case of a 5x5 keyboard it might make more sense to have a “number layer”, “symbol layer”, “function layer” (volume and brightness controls), etc.

Other techniques I’ve seen explored include combining layer keys with regular characters (e.g., the spacebar might act as the “control” key if you hold it down and press another key; otherwise if you just press down and up it acts as a regular spacebar).

Keys are boring anyway

Why pigeonhole ourselves with keyboards, devices that map button presses into “keys”? Arguably more fun devices include soundboards (which map button presses into silly noises), **stream decks** (selecting between camera angles and Twitch culture things I’m too old to understand), video game controllers, etc.

The interface of virtual key codes isn’t one we necessarily have to support. Why not instead something contextual, where physical keys are mapped directly to operations in the currently focused program? We’ve talked about **window managers** previously; perhaps a 5x5 grid of buttons to could select and move windows around a 5x5 grid on your desktop?

Thin client updates

Last month I hypothesized that maybe, **just maybe**, being able to ping Google's Taipei datacenter in 3 milliseconds would mean I could use *their* computers for my computing needs, rather than buying my own desktop hardware.

I realized there are two distinct sub-problems here:

1. What local hardware is required to hook up my monitor, keyboard, and trackpad to The Cloud?
2. What "remote desktop" software doesn't have an infuriating amount of latency?

I decided to table the first problem for now, since it's only worth figuring once after low-latency remoting works from my current local hardware (i.e., my laptop).

As for the second problem: Five of you emailed to extol the virtues of **Mosh**, the "mobile shell". But, as I said last time, there's more to life than text: I'm aiming to do *graphical computing* on the remote. (Fancy stuff, like reading PDFs.)

I'm adamant that anything short of a full graphical environment is a leaky abstraction that will inevitably lead to madness. (Shout out to the sixth newsletter reader who emailed about Mosh to say it was "a nightmare for clipboard management".)

Anyway, I **tried out more remote desktop software**. The current lead contender is **No Machine**, a proprietary solution with a generous free version. In addition to being the most functional software I've tried thus far (I can watch YouTube on the remote!), happily several factors suggest it's worth trusting:

- It's commercial — i.e., it's someone's actual job to keep it working (unlike open source).
- The business model is clear: They **charge \$\$\$** for enterprise features.
- The Lindy effect suggests it'll stick around (webpage copyright since 2002).
- Their landing page is boring.

That said, there is still a noticeable latency — my best guess is that it's caused by software video encoding on the server side. I'd like to test this

hypothesis (and also **Parsec**, a videogame-oriented remote desktop solution with a VC-alarm-bell suspiciously fun landing page), but unfortunately Google refuses to rent me a GPU due to my lack of billing history. Apparently the Cloud part of Google isn't connected to the AdSense part, where my account has spent thousands. `~_(ツ)_/`

We'll find out next month whether a timely payment of my \$1.72 bill sufficiently establishes my credibility to rent a GPU.

Works in progress

(I already keep a **list of open interests**, but thought it'd be worth mentioning here what I'm actively exploring in case any of y'all have suggestions or collaboration interest.)

Remotely controlled remote control: My apartment has mini-split air conditioners, controlled via an infrared remote. I turn them off when leaving the house, but it'd be nice to turn them back on remotely (e.g., when I am returning from lunch and would like to arrive to an optimal-napping-temperature bedroom). I'll design my own IR transmitter using the same bluetooth microcontrollers I used in the keyboard – but how should I access those remotely? Should I run some kind of **open source IoT pile of code** on a Raspberry Pi and, uh, exposed that to the Internet? Should I roll my own thing using a VPN or **the Patch Bay**? If I wanted to sell this hardware as a product to other nerds, what functionality would they be willing to pay for? (“IR Blaster” is the term of art here, and a quick Amazon search shows many existing offerings between \$30–150.)

Microcontroller superoptimization: There's a direct relationship between my keyboard's firmware efficiency and its battery life. This got me wondering about how programming languages usually force me to over-specify my intent and thus prevent certain optimizations. E.g., I can't say “read registers A, B, C into D, E, F” without specifying an order in my source code. The compiler must respect that order, even if I don't actually care about it and reading in some other order might be faster⁴.

The **Bloom** language explored these ideas in a distributed systems context, but I'm not aware of any close-to-the-machine embedded language work.

More broadly, I've been struggling with doing compile-time calculations in Rust. The macro system and const fns are complex yet deliberately limited; the sorts of calculations I'm trying to run (integer division to choose clock prescalers; assert-at-compile-time arithmetic relationships between constants) always end up being "puzzles". My solution thus far — do it in a spreadsheet and copy into the source code magic numbers with comments — is unsatisfying in the same way that using the back of a screwdriver as a hammer is both expedient yet feels *deeply wrong*. (Zig's [approach to compile-time](#) looks much more interesting and cohesive.)

In the embedded domain, such needs seem to be addressed by increasingly complex "build tools" (compiler preprocessor macros, make targets, etc.), but I can't help but wonder what a more deliberately designed system might look like. There's a lot of interesting work to draw from staged languages, [program synthesis](#), [refinement types](#), [superoptimizers](#), and [solver-aided programming](#).

I won't write a programming language, I won't write a programming language, I won't write...

Misc. Stuff

- Clojure has been the most influential single work on my personal technical growth, and it's fascinating to read Rich Hickey's [reflection and history](#) on the language. Thanks Rich!
- A [Lego submarine](#).
- "When this pandemic is "over" (whatever that means), the undead bones of neoliberal governance may well yet again gather themselves from the chaos and reconstitute the suave, smooth-talking vampire to whose predations we have grown unhappily accustomed. But they may not. We may find ourselves in a period of social experimentation and change. If so, as we diminish (not eliminate!) the role of markets, it is useful I think to understand the variety of functions that markets serve, so that framers of new institutions understand what will be excised, what may sometimes need to be replaced. So. Here are [four functions of markets](#)."

- We should all aspire to have a dentist who'll **x-ray our circuit boards**.
- **UX analysis of pockets** (and pants prototyping!)
- The surest way to a fast web site: **a public dashboard** so developers feel bad when they make a slow web site.
- "You'll never get in that package without a pair of scissors, so **don't even try**."
- **Shamir's secret sharing scheme**.
- Theory: Prevent others from reading your embedded device firmware by setting this non-volatile lock register. Practice: How can the CPU know it's locked if we **kick it when it tries to read that register**?
- "If we were feeling snarky, we could perhaps describe IPv6 as **the String Theory of networking**: a decades-long boondoggle that attracts True Believers, gets you flamed intensely if you question the doctrine, and which is notable mainly for how much progress it has held back."
- Introduction to **compressed sensing**, with examples in Python.
- "We all know what it signifies but not exactly what it means—and that was the intent." A history of the **biohazard symbol**.
- **Bitcoin is more like ham radio than the early internet**.
- Electric unicycles now go **45 mph**.
- **Seeking truth in a time of misinformation**.
- High-performance brain-to-text communication via **imagined handwriting**.
- Do you have two weeks of non-perishable food and water?

1. Yeah **TSMC** is cool, but the fried chicken and stewed pork technology here is *next level*.

2. Taiwan ended up issuing five **30-day automatic visa extensions** in response to the pandemic, starting in March going through July. I'd be curious to see a ranking of countries along this easily measurable government "competence/compassion" metric.
3. "Our police" is a broad phrasing; in the case of Portland, the city and state officials refer to the amalgamation of US Marshals, Homeland Security, and Border Patrol (!?!) officers as "federal troops". Personally, I use a simple heuristic regarding all state-armed individuals: The ones wearing high-visibility vests want you to find them so they can help you. The ones wearing camouflage think they're in a war zone and may treat you like The Enemy.
4. How could reordering register reads make a program faster? Well, on an ARM processor where both the instruction size and memory address space is 32-bits, memory reads take a few instructions: Load the lower half of address into register A and upper half into register B, then read from the address given by A and B together. So if you want to read from several registers, it would be more efficient to read ones that share an upper/lower address together, to minimize the number of times that half-address needs to be loaded into a register.

This was an issue of Kevin's newsletter; [sign up](#) or [see other issues](#).

[← Back to Kevin's homepage](#)