Ask HN: Small scripts, hacks and automations you're proud of?

216 points by ThePhysicist 16 hours ago | hide | past | favorite | 238 comments

I sometimes think back fondly of the small automation scripts I wrote over the years, so I wanted to ask you: Which scripts, hacks & automations did you create that your're proud of? Personally, I e.g. built a bot to automatically answer apartment ads [1], a small script to run Borg backup once per day [2], and a small automation to track the temperature in my aquarium [3].

1: https://gist.github.com/adewes/c9b2a71457c6c6f01f2f

2: https://gist.github.com/adewes/02e8a1f662d100a7ed80627801d0a...

3: https://gist.github.com/adewes/7a4c20a5a7379e19d78ba54521d3d...

add comment

jimt1234 5 hours ago | next [–]

Honestly, the script/hack/automation I'm still most proud of is from around 1983, when I was 12 years old, for my Commodore 64. Back then, all computer communications were over the phone line, and in my house, we only had one line, so I spent a lot of late nights, dialing into BBSs, often not going to sleep until the sun came up. Then, I would get in trouble at school for falling asleep in class. So I came up with a little program I called "Simon". I named it after a popular kids game called "Simon Says". It was basically a 1983 version of Selenium. I created instruction files, which would tell the main program (Simon) what to do: the time to dial a BBS, the number to dial, the login info, and then all the commands to send. It did things like download my messages so I could read them the next day. Send messages. But mostly, and kinda shamefully, it was used to pirate games. I had a list of about 10 local BBSs that I would dial every night, and a list of games I was looking for. If Simon found a game on the list, it downloaded the game for me. Then, it would hang up. I even created a hard-hang-up time of 7am, when my mom woke up, because if the computer was still on the line and she needed to use the phone, she would kill me! LOL

I'm still super proud of that because it was so much fun to create. I recall "releasing" the program to a handful of friends, and they all seemed to like it, but it never really went anywhere.

reply

> edngibson 1 hour ago | parent | next [–]
>
> When you were 12? Wow.
>
> reply

> moonshinefe 4 hours ago | parent | prev | next [–]
>
> that sounds sweet haha
>
> reply

ulrikrasmussen 2 hours ago | prev | next [–]

Not technically difficult, but I remember that it made me *feel* very proud: Back in high school (2004-ish) I made a small PHP script to generate a .m3u of my whole MP3 collection and I exposed this from my home PC. Since I could install Winamp on the school PCs this allowed me to stream my music collection.

Another one was a password rotation script for my university. There was an idiotic password policy that required you to change your password every 6 months via a web form, and if you tried to change it to a password you had used before you would get an error saying that the new password couldn't match the last 24 passwords you had used. This information was very useful as it allowed me to write a script that simply changed my password 24 times and then changed it back to the one I wanted.

reply

> ptman 1 hour ago | parent | next [–]
>
> They forgot to rate-limit it =P
>
> reply

vagabund 15 hours ago | prev | next [–]

So certainly not impressive, and morally ambiguous, but when chatGPT first released, I worked with it to create a python script that crawled The New Yorker sitemaps, found articles with embedded audio versions, downloaded the mp3 by reconstructing the file url using the article ID in the page source, and then renamed the file to match the page title and saved it to a well structured Google Drive directory. Now when I walk my dog, exercise, or clean, I'll put on a random article and have a much more enjoyable experience than a podcast would offer. The mp3s were publicly available so I think it's fine for personal use?

I'd never written a line of code in my life, so going through the iterative process of getting it functioning and fixing bugs was pretty invigorating. I could see where the generated code was going awry and describe it in natural language, and chatGPT would turn that into syntax and explain why it worked. Definitely a fun way to learn.

I'm sure it's ugly/inefficient, but here's the script:

https://pastebin.com/raw/DuAVAikD

reply

johtso 14 hours ago | parent | next [–]

"going through the iterative process of getting it functioning and fixing bugs was pretty invigorating."

Damn right it is! Writing scrapers is definitely the gateway drug of programming. It's such an exciting feeling to step across the boundaries from "passive user at the mercy of the UI" to "I can do anything if I can conceptualise it (and break it up into steps)".

reply

noufalibrahim 2 hours ago | root | parent | next [–]

Completely agree. It's often the first "real program" I give to my students when I teach coding and all of them love the high of doing it.

reply

photoGrant 14 hours ago | root | parent | prev | next [–]

You summed up what dragged me from "Sure I'll learn to program a notepad or todo app so I can learn, but I won't" to "I can do anything if I think about what I want first, then how to get it!"

reply

thinky_thoughts 4 hours ago | root | parent | next [–]

+1. Writing scrapers and even API calls were the step for me that went from "I'm making my computer do something" to "I'm making my computer do something that's interacting with the outside world on terms that I programmed it to. Whoa! I can do a lot of different variations of this!"

reply

UI_at_80x24 6 hours ago | root | parent | prev | next [–]

For me it was writing MIRC scripts that monitored the IRC channels I was in. So it was scrapping the content, but not nearly as elegant as what the OP mentioned.

reply

prashantsengar 15 hours ago | parent | prev | next [–]

Pretty impressive to be able to create such a script when you haven't programmed before.

I used to freelance when I learnt to code in college and I wrote many such small automation scripts for quite a few people. Looks like the AI is already taking away some jobs ;)

reply

iambateman 4 hours ago | parent | prev | next [–]

Very cool! It sounds like you could have a career in software development if you wanted one.

reply

kmoser 7 hours ago | parent | prev | next [–]

The comments are even more valuable than the code!

reply

totetsu 5 hours ago | parent | prev | next [–]

I was trying to do this same thing to download my Dalle2 history all at once. but being behind cloudflare made it a bit hard.

reply

carpdiem 5 hours ago | prev | next [–]

Most useful script I've ever written, and legitimately how I keep track of everything important in my life:

```
#! /bin/bash

function todo {
 if [ $# -eq 0 ]
  then
   vim ~/Dropbox/todo/todo_list.txt
  else
   vim "$HOME/Dropbox/todo/todo_list_$1.txt"
 fi
}
```

Sourcing this in my shell means I just have to type...

```
todo
```

To automatically get into my default todo list which is synced to dropbox (and thus accessible everywhere on mobile as well).

While, whenever I have a new project/etc come up, I can just

```
todo new_project_name
```

And get a new, nice clean file, with the same backups and multi-device accessibility.

reply

EVa5I7bHFq9mnYK 51 minutes ago | parent | next [–]

I just have a Dropbox/todo.txt file opened permanently in Notepad++, no scripts and no need to type anything.

reply

inconceivable 4 hours ago | parent | prev | next [–]

here's my pull request

```
$ cat << EOF > ~/.bash_profile

> alias lstd='ls ~/Dropbox/todo'

> EOF
```

reply

EFreethought 4 hours ago | parent | prev | next [–]

I use Org mode in Emacs, so I have a follow up:

Do you have a format in your text files to track when things are done, like a checklist?

In Org, you can give things states like "TODO", "INPROGRESS", "DONE", "CANCELLED", add checkboxes (like "[ ]" and "[x]"), have sub-tasks/lists, etc.

I have really gotten into Org, and I have found there are a few other checklist formats/programs in plain text. Just wondering how you do it.

reply

jameskraus 2 hours ago | prev | next [–]

Easily my highest utility script so far was to get an edge while picking classes. At my university there were some classes which were very competitive and there was also system where tranches of students would get access to those classes earlier (e.g. groups of honors students -> groups of seniors -> juniors, etc). I was a lowly sophomore at the time, so I basically got what was left over.

Before I could register I could see most classes I wanted to take had plenty of availability except for one which was almost full. That class would make my schedule perfect (no classes before 10am, everything ending before 4PM, good teachers), so I was ready the moment my group got access. I logged into the janky Java applet and saw that my desired class was full. Dismayed, but not giving up, I refreshed the class list to see if someone might drop during the drop-add week. Nope, still full, but there was still time for someone to change their mind and for me to slip in. I was going to need an edge.

Another peculiarity of this system was that it was definitely a weird Java app based on some type of main frame. It must have been around for a decade at least. One of the signs of its age was that you had a limited amount of time per day during which you could be logged into the system, something like 90 minutes a day.

So I did what any good hacker would: I made _really_ good use of that limited time. I wrote an AutoHotkey script which would automatically log in, traverse all the menus and attempt to sign me up for that class. It was much faster than a human, so each run barely put a dent in my logged-in time allotment. As a cherry on top, I wired it up to a python script to push notify me if it succeeded.

After letting it run for two days, some false alarms, and tweaking, I got a delightful "You're registered!" notification on my phone and found that it had successfully gotten me the class I wanted. I'm still chasing that high to this day.

reply

kkoncevicius 15 hours ago | prev | next [–]

This is more in the "hack" category, but here is a solution for bookmarking directories[1] that I am quite fond of:

```
mkdir ~/.marks/
export CDPATH=.:~/.marks/
function mark { ln -sr "$(pwd)" ~/.marks/"$1"; }
```

Then:

```
mark @name   # add bookmark called @name for the current directory
cd @name     # jump to the bookmarked location
cd @<tab>    # list all available bookmarks
```

It can list bookmarks, auto-complete, jump to sub-directories within bookmarks, all without introducing any new commands - just `cd`.

[1]: http://karolis.koncevicius.lt/posts/fast_navigation_in_the_c...

reply

gbrindisi 9 minutes ago | parent | next [–]

Nice!

I gave up on bookmarking so I now use an alias to quickly find what I am looking for with fzf and fd:

```
p () { cd $(fd $1 $HOME -d 4 -i -E Pictures -E Desktop | fzf) }
```

reply

the_fury 15 hours ago | parent | prev | next [–]

I've just spent the last few minutes playing with the bookmarks snippet. This is both elegant and immediately useful to me. Thanks for sharing!

reply

kkoncevicius 14 hours ago | root | parent | next [–]

Glad it is useful. Just a note of advice - prefix your bookmarks with a symbol that is not used to start directory names (was @ in my case).

This is useful for two things - 1) avoiding clashes with existing directories, 2) having a quick command to list all the bookmarks: cd @<tab>

reply

p4l4g4 13 hours ago | parent | prev | next [–]

Awesome snippet! I organize most of my work using something similar to the Johnny decimal system[0] and use CDPATH for quicker lookups. But I love how you build a dynamic system with a simple oneliner!

[0] https://johnnydecimal.com/

reply

meain 4 hours ago | parent | prev | next [–]

You might be interested in hash in zsh.

https://til.hashrocket.com/posts/xsavbhlrz4-shortcuts-with-h...

reply

behnamoh 3 hours ago | parent | prev | next [–]

I just use ranger and mark/bookmark directories like in Vim.

reply

mo_42 2 hours ago | prev | next [–]

I use the following tiny script to make PDFs look like they have been scanned. There are still entities that require you to physically sign a document and scan it. So far, everyone accepted PDFs like these:

```
#!/bin/sh
ROTATION=$(shuf -n 1 -e '-' '')$(shuf -n 1 -e $(seq 0.05 .5))
convert -density 150 $1 \
  -linear-stretch '1.5%x2%' \
  -rotate ${ROTATION} \
  -attenuate '0.01' \
  +noise  Multiplicative \
  -colorspace 'gray' $2
```

reply

zamadatix 6 hours ago | prev | next [–]

I don't have the script anymore but when I was an intern at a regional hospital system ~10 years ago they were running into problems with an old wireless system where after a certain global client count and AP uptime APs would stop registering clients or roams. At the start of the day everything was fine, by night some random wireless tickets were coming in, and by next morning everything was on fire and thousands of APs worth of Wi-Fi was useless. The wireless lead was sweating bullets trying to get the vendor to figure out what was wrong but the product was nearing the end of its life plus having transitioned between companies in its lifespan so support wasn't great. As a stopgap he'd stay late and reboot all of the APs from the controller at night. It'd take about 30 minutes for everything to process and the doctors/nurses would still raise hell but a scheduled blip at night was better than everything being broken in the morning.

We had great AP density so by the 2nd night I started working on an Expect script that would log in and boot 1 AP at a time per floor-building combo (multiple areas simultaneously), wait for them to come back, wait about 30 seconds, and move to the next AP on each floor. At first I did a dry run where the reboot was commented out and the "is it back yet" was delayed by a short timer to simulate a reboot delay + make sure the script would really run without a login timeout or something stupid. This would take a couple hours but it was better than taking everything down as clients under the specific AP being cycled would just roam and have a weaker signal for a minute (assuming the system had been booted the night before). We ran it the 3rd night in place of the mass reboot and watched it carefully. It went great so we left it to run every night. We had the nightly notice from an outage to a potential outage and stopped getting complaints because, except for a particularly unlucky laptop in a corner for 1 minute, nothing was really out of the ordinary. At this point we knew we were buying a brand new wireless system that fiscal year (already in the budget) so we ended up just riding the year out with this script.

In all it was a lame reboot script shorter than the table containing the lists of APs to boot 1 by 1 but it probably had more real impact than any fancier scripts/tools I made while I was there.

reply

nagonago 6 hours ago | prev | next [–]

Nothing particularly difficult or unique, but adding a simple script in crontab to download new videos from my youtube Favorites playlist has been immensely helpful in archiving over the years, as videos inevitably get blocked or deleted.

```
    yt-dlp --download-archive archive.txt --no-overwrites -o '%(playlist_index)s - (%(uploader)s) %
(title)s.%(ext)s' --format bestvideo+bestaudio --yes-playlist --ignore-errors --write-info-json
[PLAYLIST_URL]
```

Edit: if you use this, note that you should probably sort your playlist by Date added (oldest), otherwise the playlist_index doesn't work correctly. Or you can just leave out playlist_index from the filename if you don't care about it.

reply

forinti 20 minutes ago | prev | next [–]

I have a small Perl script that has been useful for the last 20 years.

It uses OLE to open Word documents and print them to PDF. Nowadays Word has a save to pdf options, but initially the script would print to postscript and then convert to PDF with Ghost.

It runs from from IIS, but over the years it has been used as a CLI tool or with Apache.

So far nothing has shown to be as reliable as this script at converting to PDF. It would be great not to depend on Word or even Windows, but there's nothing that can convert with the same level of fidelity.

reply

PaulMest 5 hours ago | prev | next [–]

I created a callbox bot using Twilio about 10 years ago. It started as something simple: when my callbox bot is called, play dial tone 9 (to signal the callbox to open the door) and then send me a text message that someone is coming up. I wasn't too worried about people abusing the system and this worked really well.

Over the years, I've made it more complex. I added a simple menu system using text-to-speech and accepting the DTMF tones for the selection. This way I could tell people to enter a secret PIN and they could provision themselves access to my lobby. If they didn't know the secret PIN, there was an option to have it patch through to my phone so I could screen the guest using normal voice and I could manually press 9 on my cell phone.

It also has an SMS interface where I can select different modes and hours of operations. So when I host a poker night, I'd txt my bot "poker" and it would it update the menu system to be Poker themed so that when guests were at the callbox they'd be greeted with poker puns. When my friend comes to visit from Austin, I'd txt "StarCraft" and he'd get to hear some dumb SC2 puns. I have 20+ different modes now for various occasions over the years. Now when people visit they get their phones out and record the greeting half-expecting something that they would want to remember or share on social media.

Sadly, my building just replaced their callbox with a newer model that does not accept secondary input. So once the call is connected to Twilio, if the user touches any button on the callbox (e.g. to enter a secret PIN), it will disconnect the call. I suppose it's now time to use some voice-to-text options to bring back the interactivity, but I suspect the lag would make the experience more frustrating than fun.

reply

ankit219 34 minutes ago | prev | next [–]

A very basic one. I use Google drive a lot. It's a pain to go to Drive link and then look for the recent files and open it. So, I wrote a Gmail sidebar add in for the same. [1]

It does two things: 1. List the files in order of last modified or last opened. This way I can list 25 latest files and go to the ones opened by me. (no folders needed, just the last opened ones).

2. With last modified date, tells me when the file was modified last. (useful if you collect data in sheets from forms or other stuff, tells you when you have a new lead). Also tells you when the colleague you share a file with has edited it or not.

I wanted to enhance it with adding more stuff like: show diff, so I know what has changed. And then, show the content from a doc right here in sidebar, because well it's easier to type a mail by looking at that.

[1]: https://github.com/ankitmaloo/gmail-sidebar-drive

reply

adrianh 15 hours ago | prev | next [–]

In 2004, the website All Music Guide did a horrible redesign. I was so irritated that I created a Firefox extension that did only one thing — it made All Music Guide a bit more useable.

To my knowledge, this was the first "site-specific" browser extension, so I wrote a blog post about it:

https://www.holovaty.com/writing/all-music-guide/

Aaron Boodman saw my post and decided to generalize the idea; he created Greasemonkey, which was a single browser extension that could aggregate site-specific JS customizations. In the years since, this idea of "user scripts" has further developed, and I think some browsers even support the concept natively.

reply

mdaniel 13 hours ago | parent | next [–]

In that case, thank you to you both for dramatically improving my web experience -- I don't know that I could stand browsing without uBlockOrigin and ViolentMonkey

I'm not aware of any browsers that support it natively, although TBH there are so many random Firefox forks that any one of them could have just bundled the extension and called it native

reply

netsharc 6 hours ago | parent | prev | next [–]

Ah, that reminds me of how I modified sites using https://www.proxomitron.info/ , it's a Windows program that would start a local HTTP proxy and you can configure browsers to use this proxy, and it would block ads. This worked really well before the web became HTTPS-crazy.

The way it would block ads is to analyze the files it was fetching for the browser and modifying found keywords, for a crude example all IMG tags with SRC containing the string ad.* would be replaced by gray images.

I also remember using it to "fix" a redesign of a site.

reply

physicles 4 hours ago | prev | next [–]

Here are two scripts I now can't imagine living without:

[1] is the reason I no longer need a taskbar/dock. It's a sort of framework you can use to build a muscle-memory-friendly window switcher in Linux, similar to pinning windows to the taskbar in Windows -- map Win+1 to terminal, Win+2 to vs code, etc. If the app isn't running then it launches it; else, switch to it. (I wrote this because couldn't find a consistent and foolproof way to do this in Linux, and if you switch window managers you have to start from scratch)

(For better ergonomics, if the keyboard I'm using doesn't have a win/meta key on the right, I rebind right control to win/meta)

[2] is a timestamped log file for you, a human. Set up a cron job to launch it every hour to note what you were working on, or append lines from the terminal whenever you're chewing on a hard problem. Then, use the data to build an accurate picture of what you worked on during the last week, or grep last quarter's log to find out why you decided to use library A instead of library B. I have used this since 2018 and frequently go back to gain insight into the reasons for old decisions.

1: https://gist.github.com/cbd32/cbec9a32b32bd9e93b0d2696c71b5f...

2: https://gist.github.com/cbd32/f1ee2967ec0181b934639c30f4e68f...

reply

hawski 37 minutes ago | prev | next [–]

This one liner is the heart of my anemic SSG. The main idea is to remove generated files if the source has been removed.

```
comm -23 <(awk 'NR>1' "$DSTDIR/build-info") <(find "$SRCDIR" -name "*$EXT" -type f -printf "%P\n" | tee >(gen_index) | xargs -n1 "$0" "$SRCDIR" "$DSTDIR" | sort | tee -a "$DSTDIR/build-info.new") | (cd "$DSTDIR" && xargs rm)
```

The rest is here: https://gist.github.com/hadrianw/060944011acfcadd889d937b960...

reply

ellyagg 4 hours ago | prev | next [–]

The Apple Watch doesn't have a good built-in note-taking app, for a long while much to my annoyance. But it turns out the reminders function works great for taking voice notes. So I wrote an AppleScript (of all things) to scrape the reminders app for all the voice notes I take when I'm out. I use this every day and this little script has brought me such peace and joy.

https://gist.github.com/michaelterryio/478f5b93111f5f5fc57c0...

To practice my marketing copy, I wrote a not-for-profit one page site about how I use it:

https://drivingwithworkflowy.com/

reply

crabbone 41 minutes ago | prev | next [–]

I wrote some Emacs Lisp functions that I use a lot:

1. Toggle how function arguments are indented: either all on the same line, or all on separate lines.

2. Convert arbitrary piece of text into a literal string (i.e. with quotes).

3. Extension to Magit that calls git-grep.

4. Dired function that calls Ghostscript on multiple selected PDFs to combine them into a single PDF.

5. Magit extensions for internal Git commands added by the company I work for (they are all nonsense, but I have to deal with them and it's better to keep this nonsense to a minimum).

Stuff I don't use anymore, but used to be useful:

1. Govmomy Emacs Lisp wrapper for select functions that rendered information as Org tables (I don't need to work with VMWare anymore). Not sure if this counts as small, but it wasn't more than some 100-200 lines.

2. Emacs command to incrementally show duplicated lines.

reply

thinky_thoughts 4 hours ago | prev | next [–]

This tiny script keeps my gmail inbox clean and manageable.

It auto-archives any emails older than 2 days (you can change this obviously) unless they're starred - this changes my email experience in two ways:

1) My email inbox ONLY has emails that are either new or that I decided were relevant. I have two days to notice this.

2) When I'm scanning my email, if I see something important that I don't have time to deal with, I can star it and keep moving and know that it'll still be visible and in a manageable-size queue later.

When I travel off-grid it is a bit of a pain since I have to go through a lot of archives, but that can be addressed by just turning it off.

I also ruthlessly unsubscribe to things I've stopped reading to keep everything manageable.

https://github.com/AlexMakiJokela/Gmaid

reply

sciencesama 4 hours ago | parent | next [–]

Oh man this is cool, i would definitely want to delete certain mail after a week or so like the subscribed news or mail notifications.

reply

rnoorda 2 hours ago | prev | next [–]

I was teaching online, and it involved a significant amount of grading on the LMS (Learning Management System.) It was a pain to click on the grade entry field for each student, then click "Next". I wrote a browser extension to autoselect the proper input field, and then on Enter it would both submit the grade and load the next assignment. Saved me hours and hours while grading.

reply

cfiggers 15 hours ago | prev | next [–]

All of my favorite things are little AutoHotkey shortcuts. I have several that basically just execute a Regex replace on whatever text is currently highlighted. It's ridiculously convenient to just have those functions ready to go any time.

But I started accumulating so many of them, it got to be a problem just remembering all the ones I had and how to trigger them. I had to start finding new ways of keeping them all organized. That led to figuring out two things that I'm still really proud of and use every day: custom right-click menus and Emacs-style keychords.

The custom right-click menu is: any time I right-click while also holding down Ctrl or Alt, it opens a right-click dialogue run by AutoHotkey, not the default one in the given context. That lets me visually select what I want to happen rather than needing an arcane key combo for every thing. All of my frequently-used Regex functions are right-click menus now.

The key chords are: if I trigger a "prep" hotkey first—my most-used one is "Alt+G" for "Go-to"—it sets up a keyboard hook that listens for the next key and then dispatches to a function based on that. So, "Alt+G H" would be really easy to remember in my head as "Go to Hacker news." And it pops open a new tab in my browser. I use that one a lot for opening desktop applications—"Alt+O C" maps in my head to "Open vs Code." Super easy to remember.

I've automated 80% of the most common navigation items at my job this way. My coworkers have seen me do it and stopped me to ask, "Wait, tell me how you did that." It looks and feels like magic.

reply

ethanpil 15 hours ago | parent | next [–]

I'm approaching your threshold of "I started accumulating so many of them, it got to be a problem just remembering all the ones I had and how to trigger them"

Can you share a repo with your "custom right-click menus and Emacs-style keychords" to save me some time?

Love hacker news!

reply

cfiggers 7 hours ago | root | parent | next [–]

Sure, here's a Gist for each one:

- Key chords: https://gist.github.com/CFiggers/ccb6bc911ee87f423f558368128...

- Right-click menus: https://gist.github.com/CFiggers/96bc37ca468909c9ec83252144c...

Let me know if you have questions. Enjoy!

reply

puika 14 hours ago | parent | prev | next [–]

Another interesting trick is to have a capslock layer (or any other "useless" key you may have, but capslock is much easier to reach obviously) with something like:

~~~
; ahk v1 code
CapsLock::
  KeyWait, CapsLock
  If (A_PriorKey="CapsLock")
    SetCapsLockState, % GetKeyState("CapsLock","T") ? "Off" : "On"
Return
#If, GetKeyState("CapsLock", "P")
  u:: ; convert selection to code in markdown by pressing CapsLock+u
    SendRaw, ````
    SendInput {Left}
  Return
  ; ...
#If
~~~

I also find myself with way too many obscure Mouse button {1|2}, Ctrl, Alt, Capslock, F{1|12}, etc. incantations I never remember and end up opening the script to search them. I thought about adding pie menus based on context but never got to it. Would you mind sharing a minimal example of your right click menu? Sounds much simpler and accessible. The keyboard hook à la vscode chords also seems pretty useful, will definitely implement that

reply

cfiggers 7 hours ago | root | parent | next [–]

Here's some Gists that show a simple example of each concept:

- Key chords: https://gist.github.com/CFiggers/ccb6bc911ee87f423f558368128...

- Right-click menus: https://gist.github.com/CFiggers/96bc37ca468909c9ec83252144c...

Let me know if you have questions. Enjoy!

reply

123pie123 14 hours ago | parent | prev | next [–]

I did something similar but using Autoit

I used the dialog box to create little 'applications' such as listing all the printers in the building (along with description) and allowing the staff to choose the printers they wanted to use (back around 2005 ish)

I had loads of these little scripts for staff to use - and some server side stuff (mainly windows) such as generating daily configs and collecting all the logs and looking for differences

I pretty much automated half my job, allowing me to do more interesting stuff

reply

mynegation 3 hours ago | prev | next [–]

There is a specific small hotel that will remain unnamed, which my partner wanted to book for a long long time. At any given moment if you check the reservations page, it is most likely booked for two years solid. It is famous enough that people book it all the time, but it has lenient cancellation policy of 100% refund if cancelled before 2 weeks. This is not really a problem for them as these vacancies get booked in a matter of hours (as I found out).

I wrote a script on top of Puppeteer that would be called from crontab on my home server, to go to the reservation page, press the buttons to flip through the calendar and extract days with vacancy (the reservation app is an SPA). When there were new vacancies it would send me a message through one of the messaging APIs about new dates available, so I could see if it aligns with our schedule and book immediately from the phone wherever I am.

That actually worked like a charm and I snatched a very convenient reservation after few days.

reply

CGamesPlay 2 hours ago | parent | next [–]

I've done this to get consulate appointments for visas as well. Was able to reduce from a 30-day wait to ~5 days.

reply

bobnamob 3 hours ago | parent | prev | next [–]

Was the hotel in NZ? And do you work on a three headed beast?

reply

mynegation 3 hours ago | root | parent | next [–]

No, and - not sure I understand the question - but most likely, also no.

reply

lemming 3 hours ago | root | parent | next [–]

I think it's a reference to New Zealand's Managed Isolation and Quarantine system during the pandemic. Demand was far higher than availability (and quarantine was mandatory to return home from overseas), so at some point someone wrote a script to book rather than just hitting refresh. Soon, everyone was using it and they were forced to move to a lottery system instead.

reply

bobnamob 3 hours ago | root | parent | prev | next [–]

A colleague of mine built almost the exact same hack to reserve a hotel in nz mid last year, I was wondering if you were him

reply

richardw 6 hours ago | prev | next [–]

On Windows:

- Created a folder called "shortcuts"

- Copy the folder URL into the system path (Control Panel, Environment variables something something)

- Create a shortcut of anything I want to hyperlink to and put in there, with descriptive names

- In IE, I think the setting was to enable autocomplete. For some reason that was linked to the autocomplete of the run menu item in Windows.

- For extra bonus points, add a shortcut of the shortcuts folder and put in the folder.

Result is that you can hit Win-R, type e.g. "sho" and it would autocomplete "shortcuts", or "ti" for "timesheet" etc etc. Way faster than clicking around for icons, just a half-second until your oft-used app/spreadsheet/folder/website is loading.

Question: how can I do this on Mac?

reply

Pyrodogg 0 minutes ago | parent | next [–]

This shortcut tip combined with a short power shell scripts save me what feels like hours a month. Super quick access to certain things, dynamically, as the script takes input after it's name versus navigating guis.

reply

bckr 3 hours ago | parent | prev | next [–]

Re: Mac — Hammerspoon?

reply

richardw 2 hours ago | root | parent | next [–]

Looks very interesting, thanks! Apple noob here.

reply

dbalatero 7 hours ago | prev | next [–]

I made a CLI tool for quickly starting new branches with draft PRs, asking for reviews, etc: https://github.com/dbalatero/work-cli

Made a bunch of Hammerspoon plugins for macOS:

- https://github.com/dbalatero/SkyRocket.spoon - mod+click to resize/drag windows

- https://github.com/dbalatero/VimMode.spoon - Vim mode everywhere

- https://github.com/dbalatero/HyperKey.spoon - pop-up menu for when you forget your keybinds (like neovim's which-key or spacemacs)

reply

rftyuikdjhgvcf 3 hours ago | parent | next [–]

oh I've been down the "I can just fix macOS" road. The sooner you move away the better. so much time wasted on plugins like these.

you effort will just be null the second apple change the usability apis enabling hammerspoon, like they did countless times before. I don't even rename the name of the fallen apps before it.

instead now I contribute things as long lived proper features on KDE and hopefully advance humanity instead of apple profits.

reply

kanyethegreat 6 hours ago | parent | prev | next [–]

that first plugin is genius! i'm totally gonna give it a go tomorrow!

edit: just opened the second. even better!

reply

schappim 11 hours ago | prev | next [–]

I ran an e-commerce business that used Australia Post for shipping. A few years ago, I was responsible for 90% of Australia Post's lost parcel/SLA refund customer service volume in one day. Before COVID, Australia Post had a "Next Day Guarantee" for parcels sent through their Express Post network. If the parcels didn't arrive the next day, customers could choose to get a prepaid satchel or their money back.

To take advantage of this, I wrote a script that tracked our shipments from the previous 12-months and identified which parcels were late. It then lodged a new inquiry through Australia Post's website. I received a call from the head of Australia Post's customer service team, telling me I was responsible for 90% of their volume. Apparently, what was causing delays was the need to wait for us to select "get money back" or "prepaid satchel", and she asked if they could do one or the other for all inquiries. This would have impacted the queue size (guessing one of their KPIs).

In the end, we received several thousand dollars' worth of postage refunds.

reply

Aeolun 1 hour ago | parent | next [–]

I'm not sure I understand. You submitted one year of late parcel requests in one go (presumably with the script), and then you had to manually select to 'get money back' or 'prepaid satchel'?

reply

teaearlgraycold 3 hours ago | prev | next [–]

I made a script that checks CPU load of a remote machine and modulates a looping fan noise on my laptop to give that authentic heavy load experience when running remote builds :D

reply

13of40 14 hours ago | prev | next [–]

If you add a .CMD file as the first entry in a .ZIP, with compression set to "none", you can rename and run the .ZIP as a .CMD, because the PK header will be treated as a nonexistent command and execution will continue after that. With a little imagination and some PowerShell I've made that into a self-extracting archive. It's a dirty hack, but I've had it working in production for years now. I want to see if I can make the batch file chimeric so it can run in a linux shell as well. Maybe that'll to be my lazy Sunday afternoon?

reply

mdaniel 13 hours ago | parent | next [–]

> I can make the batch file chimeric so it can run in a linux shell as well. Maybe that'll to be my lazy Sunday afternoon?

You may want to draw inspiration from the latest incarnation of JetBrains build bootstrapping script:

https://github.com/JetBrains/intellij-community/blob/idea/22...

reply

nigrioid 12 hours ago | parent | prev | next [–]

Really neat.

You've probably already seen them, but redbean and APE take a similar idea further:

https://redbean.dev/

https://justine.lol/ape.html

reply

sebazzz 12 hours ago | parent | prev | next [–]

If you run a "cls" as first command in the batch file no one would notice the invalid command of the PK header either (and not get possibly confused).

reply

13of40 7 hours ago | root | parent | next [–]

Right now I'm just using the PowerShell part to move the cursor up and to the left and delete it. You can still see the PK$#@($!!!1 for a nanosecond, but that just gives it character in my opinion.

reply

tupolef 7 hours ago | prev | next [–]

This is a set of alias/functions that I use to setup tmux and my ssh connections, I'm always in tmux be it locally or in ssh, it survive session disconnections.

```
# full terminal page of \n
pgdown () {
  printf '\n%.0s' $(eval echo {1..$(( $(tput lines) - 1 ))})
}

# clear teminal and TMUX history and get to the bottom line
c () {
  echo -en "\ec"; pgdown
  if [[ -n $TMUX_PANE ]]; then
    tmux clear-history -t $TMUX_PANE
  fi
}

# force MOTD display in a new ssh session
alias sshmotd='ssh -o SetEnv=SSH_MOTD=1'

if command -v tmux 1> /dev/null; then
  # open a new tmux session named default or attach to it
  tmuxa () {
    systemd-run -q --scope --user tmux new-session -A -s default "$@"
    exit
  }

  # open a new tmux session named ssh_tmux or attach to it
```

```
    alias tmuxs='tmux new-session -A -s ssh_tmux && exit'

    # run tmux if TMUX variable is null locally or in an ssh session
    if [[ "${TERM-}" != tmux* ]] && [[ -z "${TMUX}" ]] && [[ -z "${SSH_CONNECTION}" ]]; then
      tmuxa
    elif [[ -z "${TMUX}" ]] && [[ -n "${SSH_CONNECTION}" ]]; then
      tmuxs
    fi
  fi

  # run MOTD at login only in an ssh session
  if [[ -n "${TMUX}" ]] && [[ -n "${SSH_CONNECTION}" ]] && [[ "${SSH_MOTD-}" == 1 ]] && [[ -f
"/etc/update-motd.d/00-motdfetch" ]]; then
    motd
  else
    pgdown
  fi
```

reply

garyrob 2 hours ago | prev | next [–]

in 1995, I thought that it would be good if advertisements could be served from hubs to large numbers of web sites. That would mean that advertisers wouldn't have to hire ad space salespeople, which was far too expensive for most sites. But then there was the problem of how to decide which ads to show to which visitors to those sites. To do that, it seemed like it would be a good idea to base that decision on the sites people tend to visit, which seemed like it would maximize the probability that they'd actually be interested in the ads.

I looked for a way to do that. Browser cookies seemed like a possibility, until I learned that one site couldn't read another site's cookies. When I asked people if there was any workaround I was told that there wasn't. But then I figured out how to use them to track people anyway. That hack is my answer to the OP.

I filed a patent in 1995 that showed how to do it. The patent tied that technique to my specific mathematical technique for selecting the ads the user would most likely be interested in, so it could not be used to restrict other people from using cookies that way as long as they didn't also use my specific math. The patent also contained a lot of features about how to give users complete control over their information and that protected their privacy. (That part didn't get implemented much!)

I eventually sold the patent to DoubleClick, and DoubleClick was eventually sold to Google for an amount I recall to be in excess of $1 billion. By then I'd seen a marketing site recommend that Google buy DoubleClick, with my patent listed as one of the reasons they should. DoubleClick had their own patent describing the tracking cookie, but their priority date was a couple months later than mine. And it didn't have the privacy features mine did, or the same degree of tech about how to actually select ads.

In defending against a patent troll in 2021, Google and Twitter together filed a joint brief calling the tracking cookie "Robinson's Cookie". My name is Gary Robinson. (In case anyone is curious there's more info at https://www.garyrobinson.net/2021/07/did-i-invent-browser-co....)

reply

> jonwinstanley 2 hours ago | parent | next [–]
>
> Wow. Or, is this a wind up?
>
> reply

sgarland 7 hours ago | prev | next [–]

One of my servers is an older Supermicro X11, and I've flashed its BIOS to be able to boot from an NVMe drive in a PCIe adapter. While this normally works fine (I had another identical board it worked flawlessly on), this particular one for some reason doesn't always see the drive. Since the server is off most of the time, but wakes up daily to ingest ZFS snapshots from the NAS, the ability to turn on and off reliably is important.

I didn't feel like doing that deep of a dive to determine why it sometimes sees the NVMe drive, so instead I wrote a script [0] that uses IPMI to power cycle it until it can get an ssh connection. Originally I was sending a magic packet, but realized that was only a one-shot, so I switched to calling the IPMI executable via subprocess. No idea why I left the other stuff in there.

Anyway, this has reliably worked for well over a year at this point, dutifully restarting the server until it comes up. Some days it takes 1 attempt, some days it takes 5, but it always comes up.

[0]: https://gist.github.com/stephanGarland/93c02385e344d8b338aab...

reply

asicsp 5 hours ago | prev | next [–]

Parse `man` and `help` pages to get descriptions for options: https://github.com/learnbyexample/command_help

```
    $ ch ls -vx
       ls - list directory contents

       -v     natural sort of (version) numbers within text

       -x     list entries by lines instead of by columns
```

Script to provide `cut`-like syntax for a few `awk` features: https://github.com/learnbyexample/regexp-cut

```
    $ echo '   one   two three' | rcut -f2
    two
    $ echo 'sample42string123with777numbers89' | rcut -d'[0-9]+' -f3
    with
```

reply

StopHammoTime 7 hours ago | prev | next [–]

I don’t have it on me but I worked for a company that was manually generating Let’s Encrypt Certs (about 20-30 of them) and manually deploying them every 90 days.

I automated the renewal process using TF and integrated it as much as possible with other services. I then integrated it into a CI/CD pipeline. It took DAYS to do these renewals (ironically of course, $30 per year for a fixed certificate would have been cheaper than the people time but #startuplife). It became a 30 minute process. It only took me two days of interrupted time to build this solution and get it working.

Still going after two years so it’s already saved about 900% on the labour costs.

reply

kagaw 6 hours ago | parent | next [–]

what's TF?

reply

alex_sf 6 hours ago | root | parent | next [–]

Terraform, most likely.

reply

dewey 4 hours ago | prev | next [–]

A small workflow that I come back to all the time is generating curl commands with dynamic values through SQL queries to run something against our API. People don't always think about that so once you have it in your tool belt it's a great tool you'll reach to often.

Something like the following, then just copy it in a shell script, chmod +x it and execute it:

```
    select 'curl -XPOST localhost:9090/apis/some-service/' || p.technical_name || '/locale/' ||
l.full_locale
```

reply

chrisMyzel 59 minutes ago | prev | next [–]

Don't have the code around but using a cheap NFC sticker at multiple places in my house to pause/ resume audio. I'm using a RPI with mopidy and made a small microservice that does the pause/resume when triggered. Super handy when I want to leave the apartment and come back to my music.

A GET request (by the phones browser) is sent when I "touch" the phone to any of the stickers, the backend is a simple nodeJS express script that listens for GET request on a route.

reply

testmasterflex 1 hour ago | prev | next [–]

Bought one of these on Swedish Craigslist: https://www.walmart.com/ip/Star-Micronics-TSP654IIU-24-GRY-S...

Which was the same they used at my university in the cafeteria. Managed to plagiarise the same receipt we got when ordering food (which you needed to show the kitchen to get a plate of food) and was thus able to eat for free for the rest of my enrollment.

Not even my professors could teach me to do GUIs in Java but this thing sure was ‘Swinging’

reply

liampulles 47 minutes ago | prev | next [–]

I have loads of DVDs and a hobby of mine is to back them up to my pc. So I created a utility to rip each title on a DVD to a folder, and then a CLI tool with machine learning to categorize title folders recursively under "Extra", "Episode", etc.

Bit more than a hack, but a simple thing which has made the whole process much easier

reply

nicbou 1 hour ago | prev | next [–]

I built a script that finds appointments at the citizens' office in Berlin. At first, it just collected stats. That was to test a theory that you have to check early in the morning (turn out it's false).

Finding an appointment is notoriously difficult, and people often depend on the results of those appointments to start their life here. So I built a UI for the script and put it online. Not just code on GitHub but a live website everyone can use. It uses websockets to wait for appointments from the script running on a server.

It's been live for almost a year now. Thousands of people got an appointment that way. I got the city's blessing to keep it online, albeit with a restrictive poll rate.

It's really cool to see people recommend the tool to each other, and to work directly with the city.

When I return from vacation, I want to ask the city to extend the tool, and also translate it to other languages.

reply

culturestate 1 hour ago | parent | next [–]

Thanks so much for this tool; it saved me from untold frustration when I moved to Germany. I’m happy to see that the city blessed it.

reply

nicbou 1 hour ago | root | parent | next [–]

Thank you for the kind words! We met once, and it was interesting to hear why appointments are so broken. I left them with a deep sympathy for their situation.

A good example is that people book multiple appointments and don't show up. Over 15% of appointments are no-show. However they can't ban people or set up any barriers that would disadvantage citizens.

That being said, they're hard to work with, simply due to being too busy to respond. It's frustrating.

But there are some encouraging developments happening. The creator of the booking system was put in a position to change things. We're supposed to meet soon, and I'm excited to see what will come out of it.

reply

laserbeam 2 hours ago | prev | next [–]

I used to have a script that would turn my desktop into a "whiteboard for temporary files". Every file older than 2 days would get copied to some temporary trash folder, and would then be deleted after 90 days of inactivity. I had no shortcuts on my desktop, only files and downloads. Everything there was meant to be temporary, quick sketches. Was super happy with that workflow and it was all a couple dozen lines of bash/powershell (as I migrated between OSes).

reply

nlnn 3 hours ago | prev | next [–]

One sillier one I look back fondly on was an undersea themed email signature generator I wrote maybe 20 years ago or so.

I think it was a Perl script connected to a named pipe or similar, so whenever my mutt client read it I got a new sig.

It generated 4 lines of an ASCII art underwater scene, containing a randomised selection of different fish, sharks, boats, seaweed, crabs, rocks, shells, treasure chests, etc.

There was a bunch of code to ensure no overlap, and more natural looking scenes.

There were also certain things that only triggered on certain special days of the year (Christmas etc.), and others that only appeared in conversations with people I'd exchanged more than N emails with.

Sadly it went away when my email client moved to be primarily web based.

reply

cookie_monsta 6 hours ago | prev | next [–]

In my student days I bought and sold a lot of stuff online on a marketplace website that I guess was like Craig's list. The default sorting was by "recently posted" and your ad would very quickly get bumped off the front page. You couldn't post duplicate ads, so I made a Python scraper that would download all my ad text and photos, delete the ads, then recreate them, which meant I could stay on top of the page whenever I ran the script (which eventually I ended up running as a cron job of course).

Dunno if I'm proud of it, but I sold a bunch of stuff, most of which I'd bought by having other scrapers on other sites alert me when ads meeting my criteria were posted.

reply

pdimitar 6 hours ago | prev | next [–]

I have started making bash scripts for converting various backup snapshots from one tool to another. So far I've covered:

- Borg to Restic - Borg to Kopia

They work quite fine, though Restic and Kopia lack the ability to change the root directory of the backup so if your script extracts to a unique temporary directory for each snapshot then you got some confusing output about no files being cached (Kopia) even though it technically deduplicates them quite well. (I suspect Borg doesn't have that ability as well.)

Haven't published them yet but I think I will sometime this year. If somebody pings me I'm very likely to work on them a bit more and publish them though.

reply

aborsy 4 hours ago | parent | next [–]

Since you have used several backup tools, I wonder which one you recommend.

reply

pdimitar 3 hours ago | root | parent | next [–]

Definitely Kopia. I was skeptical at first, it's a relatively new tool and not much is known about it (at least on HN) but it's uber-fast and compresses extremely well. Detection of duplicate blocks seems both more aggressive *and* faster compared to Borg and Restic.

It's honestly a win on all fronts. It can sync to remote places -- which I don't care about as I have a local directory where it does its stuff and I then distribute that with `rclone` to several encrypted cloud accounts -- BUT many people do care so to them it's an even bigger win.

So yeah, I tried Borg and Restic and Kopia and Kopia wins by all accounts: size of backup directory and speed of operation at least.

(One thing I should mention is that Kopia is also very aggressive at eliminating duplicate snapshots and sometimes you might find yourself lacking snapshots for e.g. five days as it happened with me; however, upon very close inspection -- think unpacking the two snapshots before and after that period, and checking diffs, it turned out that I lost nothing, everything I wanted was still there. But, it's a disclaimer one way or another. Plus, its policy for retaining stuff is not like that of e.g. Borg. Make sure to read up a bit on it before using. Again, it never loses data but it has a different opinion on what is "the last 10 hourly snapshots" compared to Borg.)

reply

aborsy 1 hour ago | root | parent | next [–]

Thanks a lot for sharing.

I have been very happy with restic. It has worked flawlessly. The only thing missing from what I rather to have is error correction and asymmetric encryption.

I might look into Kopia when I have time. I was a bit worried that it’s newer and might be less reliable.

reply

ptman 32 minutes ago | root | parent | prev | next [–]

What's your opinion on bupstash?

reply

pdimitar 29 minutes ago | root | parent | next [–]

Tried it again about two months ago. Even though it definitely progressed feature-wise since my last try (>1 year ago), it compresses poorly and is fairly slow. I'd definitely rank it last if you put it next to Borg, Restic and Kopia.

reply

redman25 5 hours ago | prev | next [–]

On any system I'm working on I add a stupid simple script named `can` that moves items to the trash can. I can't say how many times it has saved my bacon avoiding accidental permanent deletions over using `rm`.

reply

pncnmnp 5 hours ago | parent | next [–]

Really cool idea. For macOS, I just found out about trash https://formulae.brew.sh/formula/trash

reply

d5ve 14 hours ago | prev | next [–]

```
# Sort list by most common terms
alias sorn='sort | uniq -c | sort -n'
# Most common IP addresses:
$ cat access_log | awk '{ print $2 }' | sorn

# Last 30 modified files.
function new() {
    ls -lat "$@" | head -30 ;
}
# I just downloaded something, where is it?
$ new ~/Downloads
```

reply

jtwaleson 3 hours ago | prev | next [–]

I got annoyed with unlocking my ebike with an app every day, so I created a car key enclosure with a Bluetooth chip to unlock it. So much work! Creating a custom PCB, writing software for the nRF52 chip, learning about BLE, security levels, GATT. Deep sleep and ultra low power usage. I use it every day and am really happy with it.

reply

thehta 2 hours ago | parent | next [–]

Sounds really cool, do you have a write-up of it anywhere?

reply

jtwaleson 40 minutes ago | root | parent | next [–]

https://blog.waleson.com/2022/03/the-worlds-first-cowboy-key...

Source code here: https://github.com/jtwaleson/cowboy-bike-remote-ble

I’m currently working on finalizing the PCB version, the first one had some flaws so the battery drained really quickly.

reply

raldi 2 hours ago | prev | next [–]

Back in the modem days, I tapped into the pins between my modem and computer’s serial port and put a speaker between RX and GND, with a switch of course so I could turn it on and off.

When on, it would make data noises at a reasonable volume indicating the current bandwidth usage. If a download got stalled, it was obvious even if I didn’t have that window in the foreground, or wasn’t even sitting at the computer.

Also, different kinds of data had different sounds and cadences, and I got used to many of them. It was fun to hear an email coming in before my sound card officially announced it.

reply

sokoloff 1 hour ago | prev | next [–]

Had an ancient boiler in a poorly insulated house that would sometimes trip out until manually reset (flip the power off and back on), yielding a cold house and unhappy spouse.

Built a microcontroller driving a normally-closed relay to interrupt the power for 1 minute out of every 120. No more cold mornings. I later added temperature sensing, smarter resetting based on temperature (to prevent power cycling a working boiler), and a webserver to it: https://imgur.io/a/VM7nD74 (the graph is an SVG; everything was generated on the local ESP8266)

reply

chaps 3 hours ago | prev | next [–]

Wrote a tweet-fitting one-liner that downloads every csv-like file from every government open data portal that's public on Socrata. Gave me a fun reason to play with FIFO files to do makeshift recursion.

```
  mkfifo f;echo \0>f&xargs -E, -a<(cat f) -I{} bash -c "wget http://api.us.socrata.com/api/catalog/v1?
scroll_id={} -O-|jq -r
'.results[]|.resource.id+"\",\"\"+.metadata.domain+"\"/api/views/\""+.resource.id+"\"/rows.csv\""'|tee
<(cut -d, -f2|xargs wget -P'{}' --content-disposition)|sed q"|tee f
```

reply

nergal 1 hour ago | prev | next [–]

I've made 2 projects that I use everyday for several years now. Not sure if I'm proud really, but they are such useful tools in my daily life so I guess I should be!

One is a RSS feed reader (GORSS) for the terminal that I use to always be up to date with stuff that interests me. The other is a simple todo-list that I use for work, shopping etc (DoIT).

https://github.com/lallassu/gorss https://github.com/lallassu/doit

reply

bspammer 1 hour ago | prev | next [–]

It’s a fish function but would be easy enough to translate to bash. It opens my diary, with an optional day offset. It creates the entries with a nice file structure so it’s easy to browse without the command too. I have years of memories captured that I otherwise wouldn’t have bothered with thanks to this script.

`diary` opens the entry for today

`diary -1` opens the entry for yesterday

`diary -365` opens the entry for a year ago

https://gist.github.com/bspammer/12d6b4a8049c69c832df7c4fd1d...

reply

digitalsushi 15 hours ago | prev | next [–]

In terms of shell aliases and helper functions, I actually go out of my way to not allow myself any.

As a young person, I was on a trip to visit another ISP to learn how a VPN would work between our two companies. Their administrator, a wonderful person with 30 years of experience in the telco-turned-ISP world, had all these incredible aliases on her workstation for managing her fleet. But on the new vendor's system, a fresh install of the same OS left her almost unable to perform.

My jobs have all had the same opportunity for the same experience, and I've tried to leave it alone. I spend maybe 2/3 of my time logged into someone else's server, and I don't doubt myself, even knowing fully well it's a personal preference above all.

reply

> the_fury 14 hours ago | parent | next [–]
>
> I feel much the same way, and for much the same reason. I do, however, have a largish fleet of aliases that I've built up over the years.
>
> The vast majority of them are simple short forms. For example `tf` for `terraform` and `tfp` for `terraform plan`. My brain just naturally thknks of the full command when using the short form and, because of this, I'm not at a loss if I'm logged in somewhere without my aliases.
>
> reply

wantguns 6 hours ago | prev | next [–]

back in the covid period, i wrote a small set of scripts to automatically attend my university lectures on zoom:

https://github.com/WantGuns/auto_meet

it does the bare minimum of:

```
    - parsing calendar events (which have meet links)
    - open zoom meets on lecture's starting time
    - exit meet meets on lecture's ending time
    - shutdown machine when all the meetings were over
```

my sleep schedule was f-ed up to say the least. these scripts helped me maintain my attendance :)

reply

alexthehurst 1 hour ago | prev | next [–]

I come across quips and insights that I want to remind myself of; including them in a random message of the day as a ZSH login message is the right amount of "flair" for me. But this is too noisy if it's a new random choice each time I open a shell. Using today's date as a source of randomness gives us a random choice that's the same all day long. So this is in my .zshrc:

```
    motd() {
        date +%D > /tmp/motd_seed
        sort -R --random-source=/tmp/motd_seed ~/Dropbox/motd.txt | head -n 1
    }

    motd
```

\`\`\`

reply

patrulek 1 hour ago | prev | next [–]

When i was still onboarding at my first (junior) c++ job i organized some ruby scripts for running tests. It wasn't a rocket science, but it seems that team didn't really cared about copy-pasting bunch of code to write new tests. Im proud of this because they found it useful.

reply

tbillington 7 hours ago | prev | next [–]

This was a recursive `rm -rf node_modules` that turned into a rust learning project and got out of hand :) now supports 14 different languages/project types. https://github.com/tbillington/kondo.

reply

0x00101010 3 hours ago | prev | next [–]

Treat git directories, recursively. Is especially like the fact that I can use it together with git aliases, e.g. I use this at least once a (working) day like `gr -b /path/to/customer/git/repos fap` (--fetch --all --prune).

```
  #!/usr/bin/env bash

  set -o errexit
  set -o nounset

  usage(){
    printf "[TRACE=...] %s [-b /path/to/clones/root] (%s/w) -- <git commands/options>\n" "${0##*/}"
"$HOME"
  }

  if [[ -n "${TRACE-}" ]]; then
    set -o xtrace
  fi

  trap 'set +o xtrace' ALRM HUP INT TERM EXIT

  : "${GIT_BASE:=$HOME}"
  while getopts b:h OPT; do
    case "$OPT" in
      b) GIT_BASE="$OPTARG";;
      h) usage; exit 0;;
      *) ;;
    esac
  done
  shift $(( OPTIND - 1 ))

  while read -r REPO; do
    [[ -d "$REPO"/.git ]] || continue
    [[ "$REPO" =~ \.terraform\/modules ]] && continue

    tput setaf 8
    printf "%s … %s\n" "$REPO" "$( git -C "$REPO" config --local --get remote.origin.url )"
    tput sgr0
    if [[ $# -eq 0 ]]; then
```

```
      git -C "$REPO" status -sb
    else
      git -C "$REPO" "$@"
    fi
    printf "\n"
  done < <( fd --one-file-system --type d . "$GIT_BASE" )
```

reply

sagarpatil 5 hours ago | prev | next [–]

Problem: My kids and wife would start Air conditioner at lowest temperature (16 degrees Celsius) and forget to switch them off. At one point in time my electricity bill was 50% of my house rent.

Solution: I used Home Assistant and Frigate (local NVR) to detect if there are humans present in the room, if no humans for 10 mins, switch off everything. I actually use a combination of things to detect human presence, some of the things include motion sensor, PC active, user sleeping (10 PM to 8 AM), there's also a switch I use to turn on sleep mode if I'm taking an afternoon nap (The switch automatically turns off after 3 hours). I also use a template to measure if the AC temperature is below 22, if it is, I set it to 22 degree Celsius. The automation above saved me 15-20% on electricity bill.

reply

garyfirestorm 4 hours ago | parent | next [–]

man! i hope your wife doesnt figure this out. ``` wife walks to thermostat - set 16C, walks away. thermostat: 22C! wife pikachu face - set 16C thermostat:.... ```

reply

noufalibrahim 2 hours ago | prev | next [–]

I used to run a site for tutorial screencasts. My "video workflow" involved recording, adding an intro and outro, transcoding to multiple formats and finally uploading the videos to the Internet Archive.

I automated everything except the actual recording using a Makefile that's over here https://gist.github.com/nibrahim/2466292. I think I added some sox commands to clean up audio too in a later version but lost the file.

Another one I did was a small script to create ruling sheets for calligraphy. It's a tedious process by hand and having a script create a PDF based on nib width is a great time saver https://github.com/nibrahim/Calligraphic-Rulings

reply

ch33zer 1 hour ago | prev | next [–]

I wrote a shitty shell + python script to get me music to DJ. It would:

* Load a playlist from spotify (including fetching the API token programatically from bash... shudder) * Search youtube for the song + artist * Download the MP3 of the song from youtube * Set up the directory structure for Rekordbox DJ * I could then manually import the song

When I was too broke to buy music for DJ'ing this satisfied my craving for new music to play.

reply

cultofthecow 52 minutes ago | prev | next [–]

Apple shortcut to send document to my kindle email.

reply

salomon812 4 hours ago | prev | next [–]

I have a Python script, called t.py and a bash alias "t" that calls it. I use it for quick navigation in the terminal instead of "cd" and "pushd/popd".

The basic format is: "t <shortcut" while just "t" prints out all the shortcuts and waits for user input.

So, "t 1" goes to parent directory, faster than "cd .." Then, "t 4" goes to "cd ../../../.." And of course "t" will print out all the options.

"t a" goes to my main development git worktree, while "t b" goes to place I build from.

But the real star is the history. I embedded "`save_t_history`" into my bash prompt. That's another scropt that saves the current directory to a history file, then removes duplicates so staying in the same directory doesn't accumulate. Typing "t h" shows the history back to about 25 unique directories. Typing "t h1" goes to the last directory you were in. "t h5" goes five ago. I be working in tmux in one pane, and then jump to another and type "t h0" to bring it to the same directory.

This is basically my bread and butter for getting around.

I would post the code, but a.) it's on a work computer so technically property of my company and b.) it's a mess so I'd recommend others just build their own version.

A hint to get started is that the "t" alias is an alias for "cd `python3 /opt/t.py`"

reply

kuratkull 1 hour ago | parent | next [–]

I also have 't' as a shortcut, but 't' creates a randomly named directory in /tmp and jumps into it. 't <name>' creates a directory ~/workspaces/<name>_<datetime> and jumps into it. 'tl' always cd's into the latest directory created using 't'. Really useful if you are messing around with code and serverside stuff.

reply

BMc2020 7 hours ago | prev | next [–]

I wrote one where when the tic mark and top row number one were pressed together, it sends a single left mouse click

tic mark and top row number 2 sends a double mouse click.

It helps reduce strain on my right hand mouse clicking finger.

reply

lelanthran 3 hours ago | prev | next [–]

Maybe this already exists somewhere and I don't know about it ...

Anyway, when working on feature branches I wanted a quick way to switch to master, pull, switch back and finally rebase.

So I made a little script (https://gist.github.com/lelanthran/6c9bd1125f89e7621364878d1...) that pushes the current branch name onto a stack before switching to a new branch, and allows me to switch back to the topmost branch on the stack.

```
xgit master # Switch to master from current <really long branchname>
git pull
xgit pop # Switches back to <really long branchname>
git rebase master
```

I use this now very frequently; when I need to switch to branches I use `xgit <branch>`, because then I can switch back easily using `xgit pop`.

reply

nextlevelwizard 3 hours ago | parent | next [–]

```
g co master
g fetch
g rebase
g co -
g rebase -
```

Where ’alias g=git’ and ’g config alias.co=checkout’

reply

switz 3 hours ago | parent | prev | next [–]

it’s not a stack, but you can toggle to the prior branch with `git checkout -`

Or to move back N branches you can do: `git checkout @{-N}`

reply

warmuuh 3 hours ago | parent | prev | next [–]

You can update a branch without checking it out: git pull origin master:master Still, two commands..

reply

iancmceachern 1 hour ago | prev | next [–]

I've mapped the extra buttons on my mouse to macros for the software I use, mainly Solidworks. The most useful one is control, it allows me to select multiple items wile only using the mouse hand.

reply

harish2704 3 hours ago | prev | next [–]

Project specific Bash history.

https://github.com/harish2704/dotFiles/blob/master/home/.loc...

This script + SpaceFM file manager saving lot of time for me

reply

entitydc 7 hours ago | prev | next [–]

One of my favorite spa/hotels for a weekend retreat for my wife and I has absurd waitlists — takes a few months typically to reserve 2 nights. However, they have a 7 day cancellation policy that leads to a U shaped availability curve for rooms.

The hotel website itself is inordinately difficult to search more than 1-2 nights (each day takes 6-8 clicks to search), but it’s all queryable through an undocumented API, so I wrote a small CLI tool in Crystal that can scan the next 60-90 days (configurable, of course) to see if there have been any sudden openings.

It does it all in parallel, so I can find out within about 10 seconds if there are any rooms available within a 30 (or longer) day range.

It’s already helped booked one wedding anniversary trip and one special getaway for 2 friends. I don’t use it often, but it’s wonderful to have around.

reply

KMnO4 5 hours ago | parent | next [–]

Reminds me of a similar hack I did recently. I wanted to book tickets for a show in Vegas, and for whatever reason the “ticket bundle” which included the show plus extras was about $20 cheaper than just the show on its own.

The only problem was selecting the bundle would automatically choose the seat (on the far side which weren’t good), skipping the seat selection modal.

I realized that if you had a seat in your cart, it would reserve it for 15 minutes and the bundle would pick the next available seat. Since I wanted central seats, I wrote a simple Puppeteer script which selected the bundle in multiple browser instances, reserving all the unwanted seats until mine was finally available.

reply

rewgs 7 hours ago | parent | prev | next [–]

> but it’s all queryable through an undocumented API

How did you go about querying this?

reply

entitydc 6 hours ago | root | parent | next [–]

Mostly recording and replaying the network request logs through Chrome and comparing diffs as I clicked around, looking at headers and URL params generated from clicking to get a high level picture for how it all fit together.

They didn’t do anything fancy in terms of auth, etc, so the only thing that was challenging was guessing some additional parameters and formats for things like number of nights, etc., once I had the basic structure.

Once I had a basic search working, the rest of it was pretty straightforward. It works for other hotels that use this booking software as well, but I didn’t bother to go down that rabbit hole much further as I didn’t want to encourage adversarial techniques and I only need to use it a few times a year.

reply

jagged-chisel 6 hours ago | root | parent | prev | next [–]

REST APIs can be queried easily enough with curl or a gui like Postman. The more impressive bit is working out the API interface.

reply

netsharc 6 hours ago | root | parent | prev | next [–]

My guess is the developer tools. Maybe the web page makes you open a date dropdown for "check-in" > select check-in date, same for checkout, hit "search", and then a pop-up would open saying "sorry, nothing available", which you'd have to dismiss before repeating the process, but in the developer tools you might be a able to see that there's an XHR query which is just something like /searchAvailability?checkIn=20230313&checkOut=20230315/&guests=2, and OP's script could just modify the dates and hit this request URL.

reply

entitydc 6 hours ago | root | parent | next [–]

Yep, pretty much this along with some guessing at the stuff that was obfuscated (mostly response structure and the meaning of some of the data).

reply

quarkw 7 hours ago | prev | next [–]

I modified the multitouch code for Karabiner Elements to support a palm activated action that can be used to add another layer of keys to my MacBook’s built in keyboard

reply

unrealp 6 hours ago | prev | next [–]

Power saving on lab setups when nobody is using them. they have google calendar for booking. so a daemon looks calendar every hour, if no booking, turns off outlets (the power supplies are remotely controllable over telnet).

reply

ianbutler 5 hours ago | prev | next [–]

I'm not sure this qualifies as small since I've been working on it since last summer and has grown to over a few thousand lines of code, but it is definitely automation and something I'm quite proud of.

https://github.com/TorbFoundry/torb

"Torb is a tool for quickly setting up best practice development infrastructure on Kubernetes along with development stacks that have reasonably sane defaults. Instead of taking a couple hours to get a project started and then a week to get your infrastructure correct, do all of that in a couple minutes."

My ambition is to fill the space of Heroku and right now I can get something like a React App and a Flask API or a full Elixir Phoenix project started and running in less than 5 minutes on Kube, with a full infrastructure as code environment, reproducibility and more.

It's definitely not meeting my ambition yet, but it's definitely in a place where I think people can use it and get a lot of value from it. I've been testing it with some friends over a few months and have been dog fooding it on my other projects.

I just finished adding a file system watcher to it over the last few days so you can iterate and changes will quickly be reflected onto your cluster. Next I'm going to extract this out into a library from the CLI tool, build an API over it and offer a hosted solution and a web app.

I've been meaning to share it with people but I've been heads down building for maybe a little longer than I should have been.

reply

dzamlo 1 hour ago | prev | next [–]

It's likely my PS3000 bash prompt: https://gitlab.com/dzamlo/ps3000

I haven't updated it in a few years because it just works

reply

tomcam 15 hours ago | prev | next [–]

I write a ton of CSS and could not find a command line stylesheet checker. I did find the NuHTML checker from W3C, which also checks the file’s CSS: https://github.com/validator/validator/releases/tag/20.6.30

AFAIK it doesn’t have an option to check a standalone CSS file.

I wrote a script to insert a CSS file into a dummy one-line HTML file, so that I can pass a CSS file to the script, and when errors are generated from the CSS, they are given using the correct line number.

I have written many more complex and perhaps more interesting Scripps, but this is the one I am, by far the most proud of.

reply

> tomcam 2 hours ago | parent | next [–]
>
> * scripts not Scripps. Thanks, Siri.
>
> reply

agumonkey 2 hours ago | prev | next [–]

One of the simplest piece of code, to help a colleague from a non IT department. Excel VBA hash count to show duplicates. Their management instructed them to use a copy/paste nested formula trick which was exponentially slower. It was 1 second now. Turned her week of dread (thousands of rows) into 1h of simple work. 8 Lines of code, 6 minutes.

reply

goodburb 6 hours ago | prev | next [–]

1: Simple tank level measurement with any tasmota capable devices and ultrasonic module: https://gist.github.com/TalalMash/2538ed7371b35cb3f9e5018872...

2: Auto restart ADSL modem on connectivity loss: https://gist.github.com/TalalMash/1fdb6363b46575d05530c6f654...

reply

hprotagonist 15 hours ago | prev | next [–]

```
  drag () {
        dragon --and-exit --on-top "$@"
  }
```

I write a lot of extremely simple but handy shell functions.

This one lets me drag/and drop things out of a terminal session (kind of) into applications with https://github.com/mwh/dragon and i use it way too often!

reply

ultrahax 5 hours ago | prev | next [–]

When I worked at IBM circa ~2006 you'd get a written warning called a clean desk violation if you left your workstation unlocked.

I wrote a little daemon that'd l2ping my Nokia brick phone; if it didn't get a response for 30 seconds it'd invoke xscreensaver. Saved me a lot of paperwork.

I currently work at a Call of Duty studio. My favorite hacks (not super high tech, but the ones that had the most impact for the least code, and the ones I feel I can talk about..):

* Put together a little box that polls varied knobs on a USB midi device to mangle the traffic going across its two interfaces. Allows for real time latency / jitter / packet loss testing https://twitter.com/ultrahax/status/1200902654882242562

* Studio LAN game browser didn't work across subnet boundaries (studio is a couple of class B's). Wrote a little daemon that'd take game discovery packets from one subnet, mangle the src addr, and send it on its merry way. Everyone can see everyone's games, happy producers.

reply

notjulianjaynes 4 hours ago | prev | next [–]

I don't remember the exact script, but once had a shit temp job doing QA on standardized test software. Basically, Here's the answers, take the test and make sure it doesn't fuck up somehow. Used autohotkeys in a way that I could do 80% of my workload with only 4 keys of keyboard. (Multiple choice test). It was basically just abcd then a bunch of tabs and shift + tabs to navigate through the otherwise tedious UI. I considered bringing in a guitar hero controller and mapping them to that, but got fired for sitting on my desk before I had the chance.

Why it was even nessecary to dothis in the first place rather than a script that also entered the correct answer was presumably some weird thing in the contract with the state who administered the test.

reply

aliendpoodle 4 hours ago | prev | next [–]

I can't even remember the year this started but it was definitely what got me into coding.

Virus that turned files and folders on removable drives into hidden system files was a popular thing that antivirus progs just didn't help with was everywhere. Friend's and co-workers would constantly bombard me for help recovering the files. When I figured out essentially how the pesky virus worked (this was before the variants that made reg edits) I coded something to do the opposite in a batch script. Would then just set it to identify the portable drive added and it'd delete the virus....the .lnk shortcut files it created and then unhide and return the folders and files back to normal.

Nothing fancy but it did influence my career direction alot.

reply

franciscop 6 hours ago | prev | next [–]

I made a small tool to automatically handle git/npm stuff without worries. Think of it as a automated tiny local CI, you just run "happy" and can rest easy knowing that in few seconds/minutes (depending on your build+tests+etc) the latest running version will be deployed in git/npm/etc:

```
npm install happy -g

$ happy

// Basically does:
- npm build (if package.json build available)
- npm lint (if package.json lint available)
- npm test (if package.json test available)
- git add . && git commit -m $MESSAGE || "Saved on $DATE"
- git pull
- git push

happy --now
// Same but omit the build+lint+test steps, ideal for e.g. a documentation change

happy --major | --minor | --patch | --publish 1.2.3
// Same but use `np` (install separately) to bump the {publish} version and publish it in npm
```

It is built to maintain many small Git/npm projects, which fits my usecase and needs perfectly. But it has some limitations ofc, like it works assuming small projects working on a single git branch so if you have any kind of more advanced git setup it won't work properly.

reply

2020aj 4 hours ago | prev | next [–]

Script to pull, patch and build libinput with a modified touchpad acceleration function to be more similar to Windows touchpad acceleration. Triggered by a pacman hook whenever libinput gets updated. I think they may have merged the ability to create custom acceleration profiles now though?

reply

thih9 4 hours ago | prev | next [–]

> a bot to automatically answer apartment ads

Isn’t that spamming? The bot answers ads that you haven’t even seen and that you might reject based on description/photos. There could be other people genuinely interested in these ads who will have a harder time contacting the landlord.

reply

eternityforest 4 hours ago | prev | next [–]

I've done three worth mentioning.

One is GenFortune: https://github.com/EternityForest/GenFortune Which is like fortune, but it generates them using a MadLibs style algorithm from data files.

One is a candle flicker simulation algorithm. I did not make this demo, I wrote the original for the PIC, they adapted a version of it for JS: http://blackdice.github.io/Candle/

At the time cheap LED candles weren't very good, but it seems many are a lot better now. I used a model assuming that the flickering was wind driven, that the flame always wants to rise towards it's maximum at a certain rate, but that depending on the current wind speed at any time it can get randomly "toppled" to a lower value, while then wind settles down to a baseline but can randomly jump up in a gust to a higher speed.

Of course, now we have multicolor multipixel flame algorithms that do way better, but this is pretty good on a single pixel light.

I still use mostly the same algorithm for DMX flame effects in Python, but I apply then effect less to the red channel so it gets redder when the light goes down for a little added variety.

The other is this RNG. It does not even have a full period for it's 32 bits of state but it's very fast on PIC10 type chips. I can't think of any reason I'd use it. I was like 15. But of everything I've ever made, this gets the most attention, and I'm not entirely sure why. It doesn't even pass all statistical tests. I'd probably use it over LCG though.

``` uint8_t x,a,b,c; uint8_t randRNG8() // Originally unsigned char randomize(). { x++; //x is incremented every round and is not affected by any other variable a = (a^c^x); //note the mix of addition and XOR b = (b+a); //And the use of very few instructions c = ((c+(b>>1))^a); //the right shift is to ensure that high-order bits from b can affect return(c); //low order bits of other variables } ```

reply

jeffreygoesto 14 hours ago | prev | next [–]

Rewrote grabbing car sales ads from Selenium to Playwright and exported the data of a specific search to a spresdsheet. Could easily derive trend lines for price over mileage and price over age and since there is a nice unified JSON somewhere in each page with all the features per car, I could see if a more expensive one was worth it due to more features, or not. Served me well when buying but also selling a car.

reply

samstave 13 hours ago | parent | next [–]

WANT

reply

Aperocky 4 hours ago | prev | next [–]

A simple diary script.

`$ diary` to create/open a file for today's diary in vim: https://github.com/Aperocky/diaryman/blob/master/diaryman.sh

Or try `pip install diarycli`: https://github.com/Aperocky/diarycli, for a pip packaged python version that does the exact same thing.

I've actually kept diary and work logs, things I did not know I was capable of.

reply

mynegation 3 hours ago | parent | next [–]

As someone who wrote the shell script with pretty much the same functionality - check out mkdir with -p flag! :-)

reply

ssss11 3 hours ago | prev | next [–]

I once hacked a daily retail data polling script to simply make it carry out retries on failed polls. It succeeded about 80% of the time on retry.

This saved me on avg about an hour a day of manually remoting into retail store systems to gather sales and stock data for the central system processing.

That was a good feeling, and was so simple.

reply

ssundarraj 4 hours ago | prev | next [–]

I made a bookmarklet to open a webpage in archive.today.

https://ssundarraj.com/stuff/archive_bookmarklet.html

reply

rockwotj 3 hours ago | prev | next [–]

In college we had a class registration system that was first come first serve and enrollment started at 6am or something. I created a small script in selenium to automate this so all I had to do was have my laptop running and at 6am it would sign into the portal and grab up my classes as fast as possible.

This ended up always making sure I got into the section I wanted.

reply

ht85 5 hours ago | prev | next [–]

A useful hack I've implemented in several projects is to accept the stored password hash as a valid password.

It is generally possible to impersonate a user but there are always subtle differences in the way those sessions work.

Accepting the hash lets people with database access perform a real login if needed.

reply

LouisSayers 4 hours ago | parent | next [–]

> Accepting the hash lets people with database access perform a real login if needed.

... and if anyone managed to say do an SQL injection and retrieve said hashes, then that then would give them access to your users accounts, right?

reply

alexchantavy 2 hours ago | root | parent | next [–]

And also access to those users' other accounts if they reuse passwords (most people probably do)

reply

LouisSayers 1 hour ago | root | parent | next [–]

Knowing hashes alone wouldn't allow you to do this normally - usually you use a "salt" during the hashing process which would lead to different hashes with the same password.

Without doing the OP's "trick" however, if you somehow managed to get a hashed and salted password then it still wouldn't be enough to gain access to someone's account.

reply

xeonax 2 hours ago | prev | next [–]

I run a clipboard logger + Window Title Logger in the background. I am able to find stuff from 10 years back. Works across all apps. Want to remember something just copy it. It might not be secure practice. I regularly encrypt the rolled over logs using 7z.

reply

> iAmAPencilYo 2 hours ago | parent | next [–]
>
> Would you mind sharing which tools you use or share the scripts? That sounds like a fantastic, no brainer way of saving stuff. I'm guessing there's a lot of noise too, like when you're editing code?
>
> reply

Psychotherapist 3 hours ago | prev | next [–]

I wrote a small python script for work that checks the time I have clocked in at work, checks how long I have to work that day and then recommends me the best time to leave to get the bus. The time is available via the time management website from work and the public transport API. It works surprisingly well and has saved me a little time every day.

reply

cperciva 5 hours ago | prev | next [–]

Hacker News Daily (https://www.daemonology.net/hn-daily/) is about 100 lines of shell script, including generating all the HTML.

reply

zeagle 11 hours ago | prev | next [–]

I wrote a wrapper around brscan and my brother printer with a feeder but no duplex scan or save to network folder capability to get double sided deskewed ocr'd pdfs into paperless-ngx with the touch of a physical scan to PC button.

I'm medical and a hobbyist, not technical, so I was happy to figure this out in bash with some trial and error. I scan the fronts, it gives me 15 seconds to put the backs in the way they came out and it deinterlaces after with deskew, ocr, etc. I used it to shred most of my filing cabinet and now scan bills/letters/receipts/etc. A cron job banks up paperless-ngx offsite using borg. For me it's just slick, my spouse uses it independently (!), and helped my bookkeeping a lot.

reply

WillAdams 13 hours ago | prev | next [–]

The time that my boss promised that we could write an AppleScript for InDesign which would allow adding an index to an InDesign document --- turns out it was 4 levels deep, and when I asked about doing this on the InDesign mailing list, Olav Martin Kvern who was then Adobe's Scripting Evangelist declared that it was impossible, because only the first level had support.

After a bit of experimentation I determined that it was necessary to verify the existence of a given index entry at a given level, and that if it existed, one could add additional entries --- if it didn't exist, then it was necessary to create the level above as an index entry, and then it was possible to add the new one.

I probably have the code somewhere, but haven't done any Applescripting for a while now (kind of miss it, Powershell is nowhere as comfortable, and there doesn't exist an equivalent to Application Dictionaries).

reply

p4l4g4 15 hours ago | prev | next [–]

I work at a database heavy company as an infra engineer. From time to time we get requests from devs or customer success to run a query on all customer databases to see or certain assertions hold (e.g. If the latest update fixed a problem or estimate the impact of a change).

To automate this process, we have some ansible playbook to automate most of it: give it a query and you'll get a bunch of files with the results.

While useful, the setup and cleanup can get tidious. So, I whipped up a little shell script that creates a temporary directory and symlink it to a location somewhere in $XDG_STATE_HOME. I created a few subcommands to create the directory structure, add a query, run the query, aggregate results and destroy the state. This keeps things simple for me. Even if I forget to cleanup after myself, a reboot will make sure nothing is left behind. The symlink trick to xdg state home helps me a lot to track my state, while I work on a certain problem. It allows me to easily handle multiple request at the same time, when necessary.

reply

Tade0 14 hours ago | prev | next [–]

I've created a tool for replaying HTTP archives:

https://github.com/Tade0/emergency-poncho

There are many like it, but this one is mine.

Useful when you're a front-end developer and the backend app is not easily deployable locally and the test environment is down.

Also with it you can make a blazing-fast, browsable snapshot of JIRA.

reply

elliottkember 6 hours ago | prev | next [–]

I have one! I made a Mac application to automatically turn on my webcam light whenever the computer's webcam is activated. It's a little buggy, but I still use it every day!

https://ragnar.framer.website/

reply

mwittman 6 hours ago | prev | next [–]

imgopt - shell script for image optimizing — just a convenience / batch processing wrapper more or less. Nothing earth-shattering, just really handy for me over the past year.

https://gist.github.com/wittman/ab90f0eca5124233fa5163a32f89...

# Prerequisites: ImageOptim (desktop app) https://imageoptim.com/mac

## pngquant https://github.com/kornelski/pngquant

brew install pngquant

## oxipng https://github.com/shssoichiro/oxipng

brew install oxipng

## sips (utility provided by default in OS X)

reply

lloydatkinson 14 hours ago | prev | next [–]

I have a script listed in package.json for my site. Often I don't really care about commit messages here, not because I don't find them valuable, but because so often its from editing and publishing articles. I don't really want to think of messages all the time so often end up typing random crap.

I use this now:

```
    import { execSync } from 'child_process';

    execSync('git add -A');

    const modifiedFiles = execSync('git status --porcelain')
        .toString()
        .trim()
        .split('\n')
        .map((line) => line.replace(/^\s*[MADRCU?!]{1,2}\s+/, ''));

    const fileList = modifiedFiles.join(', ');
    const commitMessage = `Change: ${fileList}`;

    execSync(`git commit -am '${commitMessage}'`);



    "changes:summary": "tsx ./tools/changes-summary.ts",
```

reply

samwillis 14 hours ago | parent | next [–]

Ooo that's nasty... I love it!

Much better than "Changes", "Update" or "WIP".

reply

lloydatkinson 10 hours ago | root | parent | next [–]

This is exactly why I made it, in fact a lot of commits in there are wfwefewf, frefrregre, stuff, wefewfwwefew

Every time I'm tempted to figure out how to squash/collapse the entire history down to a single commit and then always use that script.

reply

StrangeATractor 7 hours ago | prev | next [–]

My first non-trivial bash script was a CLI frontend for an MPV daemon that used regex to manipulate the playlist. It also had all the basic controls you'd expect from a music player. Kind of like a CLI quod-libet. That thing was awesome.

Sadly it was lost in a when the HDD crashed and I never got around to remaking it because reasons.

reply

bhupesh 3 hours ago | prev | next [–]

Ugit (undo git): https://github.com/Bhupesh-V/ugit/blob/master/ugit

I got this idea from a Twitter discussion which eventually blew up a year later after its initial release.

reply

thastings 14 hours ago | prev | next [–]

I used to work with a cardiac electrophysiology group. The software used for acquisition and analysis was quite old, written for DOS back in the mid-1990s. At acquisition, files were named like YYMMDDXX.0nn where XX stands for the setup where the acquisition took place, and the nn in the end was the ordinal number of each file. These were encrypted binary files, so we were completely dependent on the original software.

At analysis, files were containing multiple sweeps were averaged to reduce random noise (YYMMDDXX.Ann files), then these files were exported as a CSV-like series of time-voltage paires (YYMMDDXX.Tnn files). Tables containing the calculated variables were also generated, and exported (YYMMDD.Nnn files). The fun part is that *these files had to be named MANUALLY*. Each and every one of them. I can't stress hard enough how repetitive this got... We generated double-digit number of files every day, and analyzing each file thorougly required around 30-50 keypresses to move around in the menu and to name the files. Lucky for me, no mouse use was required, and keypresses could at least be automated.

I used DOSBox on Debian to do the analysis, and I ended up creating a bash script that could automatically analyze whole folders of these files in a few minutes. To achieve this, I generated xmacro files that would be played back while the DOSBox window was opened. Opening the file was also put in these xmacro files. The generation of the files was wrapped inside a bash script that kept track both of the files in the folder and of the files generated by the analysis. If a file was supposed to be there but some something broke inside DOSBox, it would just stop playing the macro for the next file, so it could be restarted relatively easily.

A few months later, I met the guy who wrote the software for our team, and asked him if he could write us a script to unpack the binaries into CSVs. From there, I could come up with my own completely automated solution for analysis, and everything was much-much faster. I also showed him the macro-monster I created. I'm still not sure if he was amazed or he just thought that I was impatient.

reply

zora_goron 11 hours ago | prev | next [–]

I wrote a quick integration to the GPT-3 API to Emacs [0]. Although it's a very simple implementation, it's made many small things I would've previously done manually much quicker (ie - one-off calls to convert a hyperlink in X format to markdown; abbreviate first names of author lists of papers; convert some "text message speak" messages into a paragraph).

[0] https://github.com/samrawal/gpt-emacs-macro

reply

hyperific 15 hours ago | prev | next [–]

I wrote some scripts to automate the tedious parts of my photo-to-texture pipeline. I learned enough of Blender's API to make a small plug-in to handle some of the rendering tasks. None of these are particularly sophisticated or impressive but they've saved me a lot of time.

reply

olalonde 5 hours ago | prev | next [–]

Recently developed a small embedded device (using a nRF52840 microcontroller) that allows me to open my front gate remotely using my phone. I no longer have to carry around a gate remote. And my friends/guests can now open the gate from their phone as well. I couldn't find anything on the market for doing that so I'm glad I managed to do it (and it was a good opportunity to learn some embbedded programming and electronics).

reply

djsavvy 2 hours ago | parent | next [–]

Neat! How did you reverse engineer the signal that your microcontroller had to send to the gate?

reply

olalonde 35 minutes ago | root | parent | next [–]

I got pretty far at decoding the signal with a RTL-SDR radio that I bought. It was a OOK code around 433mhz if I recall correctly. But I eventually abandoned that strategy as there was a simpler way to activate the gate through a direct electrical signal. So the device is just acts as a sort of switch now.

reply

bosch_mind 6 hours ago | prev | next [–]

Nothing specific per se, but I recently retrofitted this IBM Thinkpad I bought off ebay and installed Linux Mint onto it running i3WM. I basically control the entire OS with my tiny keyboard. I have a ton of hotkeys to just map movement of everything insanely fast. Using this alongside VIM allows for very efficient usage of my machine. Highly recommend.

Aside from this, I basically just rely on things like cntrl + p for search, various bash/shell scripting + chatgpt to be efficient. Always improving!

reply

SN76477 7 hours ago | prev | next [–]

Back around 2013 I needed trello for a project.. I couldn't seem to get the UI right and decided to use gmail as the front end. It worked perfectly, and was easy for everyone to understand.

reply

justinlloyd 14 hours ago | prev | next [–]

Short little scripts and applications I am quite proud of (most available on github, most have videos on youtube): A Python script that would go through a bunch of PDFs submitted to a conference and slap on the copyright, current year, and conference name so my wife didn't have to hand-edit the fifty or sixty papers submitted, which regularly happened about 10 times per year before the pandemic.

A couple of PowerShell scripts that moves multi-gigabyte files from the computer that generates it to an intermediate server, and then another computer picks those up and puts them on to whatever storage space is available, squeezing them in via a naive backpack algorithm. It shaved about two hours per day off my workload, instead of manually copying files around the network and then monitoring them to make sure they got where they meant to go.

A couple of Python scripts that tell a number of SONY video cameras to wake up, pull in the video and audio over USB, verify the files transferred correctly, then erase them from the camera to make space on the SD card.

A simple voice-controlled personal assistant that can start & stop multiple video cameras, perform the "mechanical clapper" effect for each scene and take, label the video files, and read out the prompt for the scene about to be shot.

A simple C# stopwatch timer for Windows that logs my daily walks via a touch screen computer located near the front door, calculating speed & distance for that walk, total daily time, etc. Hooks in to a couple of security cameras to track me as I walk past the house on my laps, recognize that it is me, and time stamp my passing (by the camera).

A simple Python script that pulled a whole bunch of data out of PDFs for a research project. It saved the small research team multiple thousands of dollars that they couldn't get a research grant for, and was able to then save the department multiple tens of thousands of dollars each year from that point onward. I got paid a large pizza (toppings of my choice). And made a couple of friends of some librarians.

A home point touch off position script for my CNC.

There are dozens more I'm proud of that I am forgetting. These are the kinds of interesting little projects that take an afternoon or an evening and often have a huge impact on quality-of-life.

reply

iwalton3 12 hours ago | prev | next [–]

This bash config allows you to cd directly to the most commonly used directory with a specific name. It also has tab autocomplete. It does require logging all used directories though.

```
  tabChar=$'\t'
  function prompt_command {
      echo "$(date +%Y-%m-%d--%H-%M-%S)$tabChar$(hostname)$tabChar$PWD$tabChar$(history 1)" >>
~/.full_history
  }
  export PROMPT_COMMAND=prompt_command

  function c {
      while read -r _ dir
      do
          if [[ -e "$dir" ]]
          then
              echo "$dir"
              cd "$dir"
              break
          fi
      done < <(cat ~/.full_history | tail -n 10000 \
          | cut -f 3 | sort | uniq -dc | sort -hr | grep "/$1$")
  }

  function _c {
      local IFS=$'\n'
      COMPREPLY=( $(cat ~/.full_history | tail -n 10000 \
          | cut -f 3 | sort | uniq -dc | sort -hr \
          | sed 's/.*\///g' | grep "^$2") )
  }
  complete -F _c c
```

reply

revscat 6 hours ago | parent | next [–]

https://github.com/ajeetdsouza/zoxide

reply

kuratkull 1 hour ago | root | parent | next [–]

+1 for z. one of the best timesavers available.

reply

dparen12 12 hours ago | prev | next [–]

I made a fzf-powered tool that makes cd'ing into deep directory trees quite fast and easy. It's somewhat reminiscent of ranger/nnn/broot, but in my mind, it eliminates the mental context switch of entering/exiting those tools.

Some gifs (with speed comparisons) here: https://github.com/dp12/probe

reply

higgins 5 hours ago | prev | next [–]

`clean`

To regularly remove any files in commonly dirty directories that haven't been viewed in a week

Keeps the desktop tidy: https://gist.github.com/higgins/b825103ce0bcf3fbefc79a439212...

reply

davidpfarrell 11 hours ago | prev | next [–]

I'm really happy with this script I wrote that is a wrapper over update-alternatives that lets you make user-level (and even per-shell) changes:

https://github.com/TekWizely/my-alternatives

reply

channel_t 14 hours ago | prev | next [–]

Years ago, when I was receiving a very high volume of a predictable and pretty malicious looking spam, I wrote an ugly Python script to report the unwanted mail to the originating network owner. It worked well at the time and I think definitely made the lives of spammers more difficult. The volume of spam I receive has been much lower since then.

reply

Havoc 7 hours ago | prev | next [–]

Bunch of ansible that runs against proxmox. Makes setting up a quick VM or LXC a 30 second task

About same as GUI interface but with IAC advantages

reply

markus_zhang 7 hours ago | prev | next [–]

Just grokked a couple of hours and pushed out a script that does very limited yaml to dbt "transpiling". Basically we are moving from a yaml based Airflow solution to another with dbt support. There are a hundred of jobs to migrate so I figured some automation is needed. It doesn't cover every situation but I estimate it can save 50% of efforts at least.

reply

sebazzz 12 hours ago | prev | next [–]

I created a small script that runs every time I log into my work notebook. It starts VPN, Teams, Firefox, Windows Terminal and my other development tools. Huge time saver as I'm usually getting something to drink or brushing my teeth in the morning when I log in.

reply

> ismokedoinks 12 hours ago | parent | next [–]
>
> You're missing out on a billable hour per week! Two hours if you time all your updates correctly.
>
> reply

belkinm 2 hours ago | prev | next [–]

Script which parses my calendar and creates tasks in my GTD-like system in Todoist - so all my tasks are in one place and I should not jump between apps to get full picture of a day

reply

gitgud 9 hours ago | prev | next [–]

Have a few little scripts in my public [1] dotfiles repository. One of my favourites is

```
    git-clean-squashed
```

Which finds all squash-merged branches in the current Git repo, and warns you before deleting!

https://github.com/benwinding/dotfiles/blob/master/bin/git-d...

reply

iambateman 4 hours ago | prev | next [–]

I wrote SimplifyRecipe.com as a means of getting rid of the extra fluff of recipe websites, and it runs via a shortcut in iOS. Overall it works remarkably well!

reply

pkos98 14 hours ago | prev | next [–]

This little script in my i3 keybindings opens a new terminal with the same working directory as the currently focused terminal, pretty useful for my workflow:

```
  bindsym ctrl+$alt+t exec xdotool getactivewindow getwindowpid | xargs ps | grep "alacritty" &&
(xdotool getactivewindow getwindowpid | xargs pgrep -P | xargs pwdx | cut -d":" -f 2 | xargs alacritty -
-working-directory ) || WINIT_X11_SCALE_FACTOR=1 alacritty
```

https://github.com/pkos98/Dotfiles/blob/master/config/i3/con...

I probably scratched some parts together from the internet a couple of years ago.

reply

> akho 47 minutes ago | parent | next [–]
>
> `alacritty --working-directory $(xcwd)` would be a bit shorter :)
>
> reply
>
> zwog 4 hours ago | parent | prev | next [–]
>
> Alacritty has this function built-in, just define `SpawnNewInstance` in the config file:

```
    key_bindings:
        - { key: T,   mods: Control|Shift, action: SpawnNewInstance }
```

https://wiki.archlinux.org/title/Alacritty#Spawn_new_instanc...

reply

mburee 14 hours ago | parent | prev | next [–]

https://codeberg.org/dnkl/foot/wiki#user-content-spawning-ne...

Maybe something like this could do the same? But could also rely on that specific terminal emulator

reply

tiriplicamihai 12 hours ago | prev | next [–]

One of the shortcuts I use a lot:

``` mkcd() { if [$# != 1]; then echo "Usage: mkcd <dir>" else mkdir -p $1 && cd $1 fi } ```

reply

jes5199 5 hours ago | prev | next [–]

my favorite is a little command line helper called “o”. it’s a helper for copy and rename commands. If I want to edit a misspelled filename I can just type

o mv mispelled.txt

and it will make an interactive prompt that looks like:

mv mispelled.txt mispelled.txt_

but the last word is editable, so I can use the arrow keys to alter it in place

https://github.com/jes5199/o

reply

ricket 2 hours ago | parent | next [–]

btw there is a bash feature that you can use for this

touch a_tset_file.txt mv a_t{se,es}t_file.txt

it works for changing file extensions:

touch foo.txt mv foo.{txt,json}

or for adding or removing characters, just leave the other side of the comma empty

touch foo.txt mv foo{,bar}.txt mv foo{bar,}.txt

reply

alexthehurst 1 hour ago | root | parent | next [–]

This is a really clever use of glob expansion.

reply

WaitWhatHuh 14 hours ago | prev | next [–]

I inadvertently caused a VERY VERY large corporation that is a household name to change its TOS through one of my little automation scripts. By little, I mean a few dozen lines of code and some ingenuity.

The corp in question: Begins with V and ends with z and likes to screw customers.

All above board, approved by several levels of various teams/depts. The head of a specific dept sent me several strongly worded texts/voicemails. He was rather expressive to say the least over something that cost $0.

I could go in more detail but there are a few NDAs I signed.

reply

severine 13 hours ago | parent | next [–]

The corp in question: Begins with V and ends with z and likes to screw customers.

Typo? Maybe ends with n?

reply

rationalist 6 hours ago | parent | prev | next [–]

Why sign NDAs? If you weren't violating the then TOS, what did you get in exchange to sign them?

reply

pinkcan 13 hours ago | parent | prev | next [–]

if you really wanted to write about it, you could have just avoided referring to the corp name at all, tos and ndas, and just write about the technical details of this script. you could even have done so under a throwaway.

reply

netsharc 6 hours ago | root | parent | next [–]

But the technical details would make him identifiable?

Although I agree with you, as it is, OP is just bragging about the effect while we're looking to be impressed by the technical details of the hack.

reply

tehCorner 14 hours ago | prev | next [–]

I made a small script to get weekly recipes with certain macronutrient composition from chatgpt, then i linked the ingredient list with my supermarket’s api so every monday I get delivered everything

reply

timbeccue 13 hours ago | parent | next [–]

How have you found the quality of chatgpt’s recipes?

reply

rodrodriguez 11 hours ago | prev | next [–]

We created a bot server, a BASIC API and a WebAutomation open-source tool. https://github.com/GeneralBots/BotServer/blob/main/packages/...

reply

LeoPanthera 14 hours ago | prev | next [–]

I keep thinking about publishing it, but I have a script which takes a Raspberry Pi SD card image, extracts it, modifies it (mostly by automatically editing cmdline.txt and fstab) to be suitable for network booting, and then pushes it to my netboot server.

This way I can quickly boot any Pi in my house (and there are quite a few now) to any arbitrary OS image just by renaming a symlink on the server.

reply

ravivooda 14 hours ago | prev | next [–]

MyQ Garage Automation had issues with my account and wouldn’t send me notifications when the door is left open for a long time.

So I automated the notification system using IFTTT.

https://github.com/ravivooda/thoughts/blob/main/docs/garage_...

reply

robswc 15 hours ago | prev | next [–]

I actually just opensourced a personal project/tool I've been using often.

https://github.com/robswc/docker-essential-aliases

When deep diving into containers, I got tired of typing/tabbing out container names and docker formatting lol

Nothing revolutionary but it has saved me a ton of combined time.

reply

agent281 15 hours ago | parent | next [–]

This seems really nice! Makes me want to build my own with Gum filters, TBH.

reply

robswc 14 hours ago | root | parent | next [–]

is it this?

https://github.com/charmbracelet/gum

If so, you just introduced me to an incredible looking project ahah.

reply

agent281 13 hours ago | root | parent | next [–]

Yeah, that's it! Makes cobbling together shell scripts way easier. They have some other projects based around terminal UIs too. Take a look at their site it's gorgeous:

https://charm.sh/

reply

jdeep009 12 hours ago | prev | next [–]

Not a "small" script, but this allows me to painlessly switch proxy configurations in git/npm and windows.

https://github.com/JDeepD/SetProxy

reply

leipert 3 hours ago | prev | next [–]

git-posh, which does a git push and plays a sound bite from “Push it” from Salt-n-Pepa

reply

hakcermani 15 hours ago | prev | next [–]

.. wrote a simple script to maintain a different history per gnome-terminal. Not 100% mine, but extended it to a point where it is working great for me!

https://askubuntu.com/a/1458448/1679924

reply

js2 15 hours ago | prev | next [–]

Wrote this for myself and tossed it up on GitHub. It ended up being more popular than I expected. Years later it's still quietly doubt its job on by EdgeRouter:

https://github.com/jaysoffian/eap_proxy

reply

polivier 14 hours ago | prev | next [–]

I wrote a small Python script that automatically downloads a song from the active Firefox tab (usually a Youtube), fills in the ID3 fields of the resulting file using acoustid, and saves it in the correct folder according to the genre of the song.

reply

simonw 15 hours ago | prev | next [–]

I have a neat Hacker News scraping setup that I'm really pleased with.

The problem: I want to know when content from one of my sites is submitted to Hacker News, and keep track of the points and comments over time. I also want to be alerted when it happens.

Solution: https://github.com/simonw/scrape-hacker-news-by-domain/

This repo does a LOT of things.

It's an implementation of my Git scraping pattern - https://simonwillison.net/2020/Oct/9/git-scraping/ - in that it runs a script once an hour to check for more content.

It scrapes https://news.ycombinator.com/from?site=simonwillison.net (scraping the HTML because this particular feature isn't supported by the Hacker News API) using shot-scraper - a tool I built for command-line browser automation: https://shot-scraper.datasette.io/

The scraper works by running this JavaScript against the page and recording the resulting JSON to the Git repository: https://github.com/simonw/scrape-hacker-news-by-domain/blob/...

That solves the "monitor and record any changes" bit.

But... I want alerts when my content shows up.

I solve that using three more tools I built: https://datasette.io/ and https://datasette.io/plugins/datasette-atom and https://datasette.cloud/

This script here runs to push the latest scraped JSON to my SQLite database hosted using my in-development SaaS platform, Datasette Cloud: https://github.com/simonw/scrape-hacker-news-by-domain/blob/...

I defined this SQL view https://simon.datasette.cloud/data/hacker_news_posts_atom which shows the latest data in the format required by the datasette-atom plugin.

Which means I can subscribe to the resulting Atom feed (add .atom to that URL) in NetNewsWire and get alerted when my content shows up on Hacker News!

I wrote a bit more about how this all works here: https://simonwillison.net/2022/Dec/2/datasette-write-api/

reply

pacifika 14 hours ago | prev | next [–]

Hide Others for Ubuntu and derivatives https://gist.github.com/svandragt/d731081c660d374ddab846e7f4...

reply

nathants 11 hours ago | prev | next [–]

all my public gists[1].

my favorite is hypothesis die hard[2]. i need a sequence of actions that accomplishes a goal. assert there is no such sequence. error, here is such a sequence. ok thanks!

1. https://gist.github.com/nathants

2. https://gist.github.com/nathants/3dc397270d73fee57c86f246243...

reply

hnarayanan 14 hours ago | prev | next [–]

I have a shell script that I use to control Spotify from the command line on Macs: https://github.com/hnarayanan/shpotify :)

reply

11235813213455 14 hours ago | prev | next [–]

I keep my list here: https://caub.github.io/misc/

I'm still impressed by my connect4 and 2048 solver

reply

auct 4 hours ago | prev | next [–]

My powershell script to create new post for hugo with good file name like 2022-12-12_my-awesome-title https://gist.github.com/AucT/5e685d4bca9bce6abcfa8d797df148b...

Usage is hugonew 'my awesome new nice title'

reply

davchana 4 hours ago | prev | next [–]

I usually check weather on forecast.gov website, with these bookmarked urls:

https://forecast.weather.gov/MapClick.php?lat=40.7983&lon=-1...

The issue I had with these were that, these pages are always in desktop mode even on mobile. So i made a one page static website which takes & stores these urls in local storage, & fetches the one I click upon. On next pageload, it remebers the last loaded city & it loads it again.

All the page is in mobile-first view mode in css.

https://spa.bydav.in/weather/index.html

You might see nothing as you need to preload/populate the local storage with either manually copied urls or simply paste this json.

```
  [
  {
    "abb": "SNR",
    "full": "Sonora",
    "url": "https://forecast.weather.gov/MapClick.php?
lat=37.98&lon=-120.38&unit=0&lg=english&FcstType=json&TextType=1"
  },
  {
    "abb": "RDB",
    "full": "Red Bluff",
    "url": "https://forecast.weather.gov/MapClick.php?
lat=40.1781&lon=-122.2354&unit=0&lg=english&FcstType=json&TextType=1"
  },
  {
    "abb": "SFO",
    "full": "San Francisco",
    "url": "https://forecast.weather.gov/MapClick.php?
lat=37.7771&lon=-122.4197&unit=0&lg=english&FcstType=json&TextType=1"
  },
  {
    "abb": "ERK",
    "full": "Eureka",
    "url": "https://forecast.weather.gov/MapClick.php?
lat=40.8033&lon=-124.1595&unit=0&lg=english&FcstType=json&TextType=1"
  }
]
```

reply

xenophonf 6 hours ago | prev | next [–]

I'm nursing a small fleet of CentOS 7 servers while we finish up migrations and upgrades this year. Most of these servers need certificates, but the built-in certbot package is really brittle (if not actually broken) depending on what plugins you might need to use. So I wrote a short wrapper script that handles everything for me:

https://gist.github.com/xenophonf/893b323b99644290fad420a54c...

It keeps the custom certbot install updated, and if anything goes wrong—and only if something goes wrong—I'll get an email thanks to chronic. Same goes for the install command outputs when running the wrapper/certbot interactively. You'll only see error messages when something breaks. Otherwise, it looks exactly like certbot was invoked directly, just with a short delay while the script does the install/update in the background.

reply

mthoms 14 hours ago | prev [–]

Getting transaction data from banks is hard. Especially timely data. Luckily both my banks have the option for a notification email being sent immediately after each Visa or checking account transaction.

Using n8n I can parse those emails and put the data into a nice Airtable base (and a csv for backup). Next step, is to add GPS co-ordinates to each transaction.

Does anyone know if there is a way to run a secure and private location tracking service to my iPhone, or is my only hope to pay Apple the annual fee and do it myself?

(Yes: I wrote "secure and private location tracking" with a straight face - I'm hoping there's a solution out there)

reply

jonas-w 6 hours ago | parent | next [–]

There are a few open source ones:

https://www.traccar.org/ https://owntracks.org/ https://apps.nextcloud.com/apps/phonetrack (if you have a nextcloud)

The nextcloud one is by far the lightest on my phone battery, the other ones would often use very much battery. But i use android and YMMV.

reply

cweagans 7 hours ago | parent | prev | next [–]

There are a number of libraries on Github that appear to interact with "Find My" somehow -- reasonably sure you don't have to pay for that.

You might also consider something like https://www.tillerhq.com (or even just integrate with Plaid directly).

reply

SparkyMcUnicorn 14 hours ago | parent | prev [–]

Home Assistant will give you a good path to easily access this data by using Owntracks or GPSLogger.

reply