
^Z
Ctrl-ZINE
Issue 16 - Vol 2
the only printable LLM you need

About ^Z

Ctrl-ZINE (^Z) is a Ctrl-c.club/Smol Web collaborative zine that celebratestech and the Smol Web. Started in March 2023, it runs a monthly issue, where anyone can download a PDF version and a pre-folded PDF version for home printing. No digital format of the content is maintained on a Website whatsoever. Some of the topics within these issues range from Smol Web protocols and communities (ActivityPub, Tildeverse), Web-adjacent protocols (Gopher, Gemini), alternative forms of communication (HAM radio, IRC), snippets of code, artwork, and anything tech-related that is an expression of self.

Those who contribute to ^Z are passionate about what they share. They want what is best for Us, the citizens of the Web. With that, anyone with that same passion is welcome and encouraged to contribute to future issues. Further info can be found in the Editorial section of this issue. May the Smol Web live forever!

Editorial:

Ctrl-ZINE

ZINEHEAD Press
Submissions contact: loghead@ctrl-c.club
Comments/questions contact: loghead@ctrl-c.club

[EDITORIAL BACKLOG OF CHANGES: changed the Submissions contact to loghead@ctrl-c.club from previous Fastmail address]

Ctrl-c shouts:

Terminal-24 is this month on the server. Get familiar with the commandline, try new software, experiment, report your findings! Hosted by ~singletona082 (Iris Topic 442)

September marks a new month, which means expansion of the **Ctrl-c Book Club**! Finish the book from August and report on corresponding Iris thread, or/and sound off on what the next book should be! Hosted by ~lettuce (Iris Topic 423)

INDEX

Humane Mail Notifications for Tildeverse Users by *phosphorsghost*

I enjoy logging on to my shell account, running `mutt` or `alpine` and checking my mail, but what about notifications? Nagging email inbox notifications throughout my day are not my thing, but neither is missing someone! Instead, I take a different approach - let computers do what they do best: faithfully executing commands that serve my needs. I let the little Unix daemon check my mail once per day after dinner, and only notify me if there's something to read when I have some time to thoughtfully respond.

Here's how it works:

I created a short shell script that runs the `mail` command, counts any (N)ew or (U)nread mail, and sends a `ntfy.sh` notification if there's at least one. I deliberately omit information about the sender or subject, both for privacy and to maintain the suspense when I log in.

```bash
#!/bin/sh
# Check for unread mail using the 'mail -H' command
UNREAD_MAIL_COUNT=$(mail -H | grep -c '^[ ]*[UN]')

# Send notification if there are unread or new mails
if [ "$UNREAD_MAIL_COUNT" -gt 0 ]; then
        curl  -s -d \
                "You have $UNREAD_MAIL_COUNT new mail(s) at tilde.team!" \
                ntfy.sh/TOPIC
fi
```

I placed this script in my home directory, made it executable, and then used Cron to schedule it to run every day after dinner.

Cron is a classic tool on Unix systems to run arbitrary commands on whatever schedule you like. While many are familiar with the system file in `/etc`, as a user on a pubnix, there is likely also a user-level cron that can run your scripts. To set it up, simply run `crontab -e` and add something like this line:

```bash
35 22 * * * MAIL=$HOME/Maildir $HOME/newmail.sh >> $HOME/cronlog 2>&1
```

That's it! Only one notification per day - nothing more. I can grab a coffee, sit down and read when it works for me. It's positively humane! If I was signed up for a busy mailing list, I might have to adjust things, but I'll leave that as an exercise to the reader. I'm happy to hear suggestions, but if you send an email, it might wait until after dinner.

Get ready for hamsterization! by *~Turboblack*

Preface:
Once upon a time, the Internet was invented, and it was invented well, but then "modern" technologies appeared, which stuffed the Internet with their frameworks, sites became thicker, more complex, fat, and this is very annoying.
***

How it all happened:
I lived at a time when in the entire area there were only a few families in whose apartments there were computers, it was the second half of the 90s. It is customary to romanticize and mystify that time, retro lovers love to talk about their old computers, and how great it was for them to spend time on them then. But for some reason very few people talk about their Internet past. The Internet at that time was thin and weak, very slow, and technologies for it were optimized so that you could reach sites from any "iron", I had a modem for 19200 baud, and a bad phone line, files were downloaded at a speed of 1 kilobyte per second. Even for that time it was too slow and annoying, the per-minute phone rates were also annoying, and I got paid for it.

Since then I fell in love with the easy Internet and optimization. Later I learned to make websites (only HTML), there was no money for paid hosting, so I was content with free, the first template was not very beautiful, banal, and simple, but I liked the process itself, and this hobby still captivates me. It was an art to fit into a certain size, make the script and graphics interesting, the site readable and beautiful

I love making websites:
My hobby grew into a temporary profession, I learned to program in PHP, and made pages for money, these were the zeros, at that time many people needed websites, so I got my hand in drawing, Photoshop, PHP, and much more.

When I got decently skilled in this business, I thought - maybe I should write my own CMS? At that time, WordPress, Joomla, Drupal, and similar monsters reigned on the Internet, which were not suitable for

a business card. But there were also many enthusiasts who made their own code, but I didn't really like their approach, too crude, complicated layout, or a very complicated approach to the process as a whole.

Sometimes, to install someone else's code, it's easier to write your own… and I started writing:

What I did was exclusively for me, I didn't show my developments to anyone, sometimes I used them for commercial tasks on freelance, but the whole idea never came to me. Many years and a dozen and a half developments led to the fact that last year I already knew clearly what I wanted!

I wanted a code that would not only be super simple, easy to install, easy to understand, super easy to understand layout, backward compatibility with old browsers, and lots and lots of templates. so the HamsterCMS appeared:

HamsterCMS is a lightweight PHP script that allows you to place a business card site (portfolio) on a hosting, while the main thing in the functionality is multi-template. You can assign each page its own template, design. For example, you have 4 pages, designed as spring-summer-autumn-winter. Interesting?

Hamster is easy to set up (just copy the files to the hosting), very easy to operate (if you don't need the admin panel, you can simply upload text files to the hosting via FTP and they will appear as pages.

The Hamster layout works on the following principle: you have an HTML template, where you want to display the menu, you insert [[NAVIGATION]] into the code, and where you want the text of the pages to be displayed, you insert [[CONTENTS]] and, accordingly, the footer will be [[footer.txt]] and if you need additional blocks (for example, for widgets), you simply create a text file in the includes folder and insert it where you want the block to be hosting:

In March of this year, I showed the hamster code to one programmer who decided to expand the functionality of his hosting http://web1.0hosting.net/ , and was looking for a simple solution

for this, and for half a year now we have fruitfully cooperate, I transferred several of my sites to the hosting (I invite you too, it's simple and easy), created a hundred ready-made templates that can be selected in the hosting admin panel.

A kind of Wix that can work even under DOS, under old Internet Explorers from the late 90s, but it works fine in modern browsers and you can edit html5 sites on it. By the way, the templates work with the usual Hamster CMS, which you can find here

https://github.com/turboblack/HamsterCMS

We are gathering a community (the hosting already has more than 300 registrations) of creative people who love minimalism, small net, and old time-tested technologies.

We invite you all. ENJOY

TINYLOGS/MICRO-BLOGS of and around the Smol Web

~skf

(gemini://gemini.ctrl-c.club/~skf/tinylog/tinylog.gmi)

2024-08-03 17:30 CEST
My old Toshiba laptop have really got a new boost since I upgraded from Debian 10 to Debian 12. The overall performance is just great, the fan doesn't have to work nearly as hard now. Happy.

2024-07-19 12:10 CEST
As I do not know much about the subject I wonder: Do not cyber security companies test their code in a safe environment before sending it out to customers?

2024-07-07 08:05 CEST
Spent almost all of my computertime yesterday on reading parts of the GNU Mailutils manual. Though I don't use mail.mailutils that much I really enjoyed myself. You never know when it is comming in handy.

2024-06-26 23:34 CEST
Debians support (LTS) for Debian 10 (Buster) is coming to an end in the end of June so I feel compelled to upgrade the operating system on my old laptop. I like Debian and going to see if my laptop will manage to run Debian 12 with the Gnome desktop in a way that is satisfying. Otherwise I'll have to pick up a lightweight distro. Am in the process now of double checking that I haven't missed to backup any important stuff. Exiting times indeed. ;-)

2024-02-07 18:20 CET
Maybe my love of the command line just is nostalgia. Happy memories from the early eighties of my schools computer with terminals (Norsk Data), 8 inch floppies and Basic. I wish I had said "I want to learn more".

2024-01-05 13.00 CET
I stopped using social media a while ago because I'm too sensitive to deal with rudeness, lies and so on. Until very recently I have seen Gemini as a sheltered place which was naive of me. Hopefully the level of stupidity is kept so low that the pleasure of reading interesting gemlogs is not lost. Again, I'm naive.

TINYLOGS/MICROBLOGS (cont..)

~pgadey

(https://ctrl-c.club/~pgadey/soc/)

2024-08-20

> [12:52] Outside enjoying the patio. I've got: my current sketchbook, journal, pad of paper for correspondence, and an old journal to be indexed. I might have a problem. Not enough stationary..
>
> [09:48] I'm indexing some stuff from July, and it's weird to see the plans work out. For example, I just read "we want to get a big table for the main floor" while sitting at the big table on the main floor.

2024-07-04

> [11:18] A CSpace post Why a Blog? from 2005.
>
> A blog is a structure for casualness, and its way of organizing things liberates you all of those worries. Freed of the neurotic worry to organize, I can just write, jot, scribble, rant, or whatever. And since blog posts are all run together, I don't have to worry that one is too small for a HTML page (or just too much work once I add the necessary framing).
>
> Blog writing doesn't have to be scribbles, and a lot of the blogs that I most enjoy reading are very carefully put together. But I'm not sure I have the time and energy for that; a good part of my purpose in trying out blogging is the theory that writing something is better than writing nothing.

What is "True Tech"? by *~loghead*

Think of it - in permacomputing, collapse infomatics, and all things that concern the belief (as I do) that humans and the Earth are not so much on a collision course with...humanity, but more in the the throes of "the best remain", or "thinning herd" theory (not so much a theory), or (as we all know and abide by) "survival of the fittest".

So where does this reality come into play with STUFF? As in, the Arctic seed vault, a novel idea, though all of nature will have it's way/preference with the botanical life on this planet, as well, and The Github Arctic Code Vault (same facility?) keeping the biggest, most reliable and consistent opensource projects stored on magnetic tape, deep underground as to reference/recover those repositories should a societal, and indeed, civilized collapse occur.

But, caveats with both (the seeds and source code) persist with these methods (though worthy of practice - just not future-proof nor lacking failsafe for either). For example, ecological conditions must *exist* to carry forward some of the seeds stored within a/the seed vault, as well as the (tech) environments to continue some of the projects on the repositories therein (some, if not many, of the projects considered wouldn't have much functionality without a datacenter, for example).

So where am I (or we (humans)) with this?

Arriving (for the latter, technology - I can't speak for future botanical kingdoms) at a spot where there is a consensus, or at least strong agreement upon, which *types* of tech can/do carry forward in a world with:

- less sources of centralized electricity
- less potential for generating centralized or decentralized electricity
- less integrity and consistency in batteries (be it AAA or a 40 pound solar brick, making a GOOD battery takes money and industrialized processes)
- less physical resources for building components (e.g. phone screens, rechargable batteries,

USB port, physical controls for any/all consumer tech, long range communication components, etc.)
- less forms of data storage
- less forms of (accessible) storage outside of fungible documentation
- less longform/long-range communication protocols
- less infrastructure, in general

Now, all Computer Science is good (or, at least most of it worth a damn), but some products and protocols have more potential to sustain and contain the levels of (CS) advancements on them (be it software or hardware) than others. Example a well-functioning Raspberry Pi can carry an abundant amount of PDF and .txt documents, as well as software IMG/APK/ISO files, and maintain their integrity (both the Pi and the files) than a MacBook Air that lacks functionality if not connected to an iCloud server).

So given the (obvious) state of "just WHAT can one thing DO", and venturing beyond that, what tech and tech preservation methods, both specific and generalized, can be utilized to make the world, and the hobbyists, computer scientists, tinkerers, and the curious amongst us, in a sort of "open seed vault" of information? Not asking "what does it take for one (or others) to retain knowledge", but what hardware/software/products/protocols and processes SHOULD be carried forward (with a low (electrical) resource future in mind)?

Some come to mind:

- Languages: C, C++, Python, low-level (sounds so derogatory) Assembly (such as for RISC, ARM chipsets)
- ports and power plugs: USB-A is likely the most abundant in this respect as of 2024. USB-C has come a fairly long way, many products adopt it, many more of the past (both in product number and sheer volume OF those products) utilize USB-A - some still being produced using USB-A exclusively (note: both USB-A male and female should be considered significantly in this regard, as things such as wall warts (wall plugs) would be less beneficial for an environment lacking a centralized power source)

- power and pixels: I query to myself "what IS the easiest way to make a battery? Not just *A* battery, but one that can deliver decent voltage and be sourced from local/accessible materials?" And keeping

the displays and digits in front of us WITH those batteries, one would consider compromise there, as well. No 1080p, HD, or even 360p displays, but opting for "bulky" pixels and display production protocols (e.g being able to build, and keep building) displays that "get the job done", albeit color, clarity, and other niceties are possible down the line
- networking: LoRa is a promising technology of broadcasting small amounts of data short and long range, should they be implemented properly. Mesh networks and "sneakernets" are always worth consideration for longevity in carrying the CS torch forward.

I consider this a Phase 1 of this document - as specifics and nuance are going to take wider audience to distill down and consolidate "what stays, what goes".

Share this document, download it, print it, mail it to someone (e-mail or snail), make a poster - hell, distill down THIS documents and put into practice/research what can be done to make it (the document) and considerations within more useful and helpful.

Vinland Saga: Breaking Free from Causality by *~Mindhunter*

It is always quite difficult to articulate why it is that a work of art leaves a lasting impression on those who witness it. Although you can try to explain such an abiding influence by deconstructing its various elements some of which in this case happens to be the extremely well woven characters, the handling of the complex dynamic between them, the amount of precision dedicated towards their history and most importantly the breathtaking story we see unfold between a son and a father. Vinland Saga is definitely impressive for all these reasons but I believe there comes a point where it becomes more than just a cocktail of greatness.

The narrative, as it unravels, places Thorfinn amidst an especially violent setting. Being the son of a great warrior and raised in the throes of consuming revenge alongside his father's murderer, Thorfinn spends most of his formative years learning how to kill. His unwavering persistence in avenging his father is at the center of a metamorphosis which transforms Thorfinn from a naive curious little kid whose eyes gleam with awe and wonder into a bloodthirsty creature lusting for vengeance. As much as it is heartbreaking to see Thorfinn become all that his father did not want him to be, it paradoxically also forms the prerequisite for understanding a rather cryptic sentence. "You have no enemies"

In a sense, Vinland Saga could be described as simply a rumination of what this truly means. A rumination that portrays the initial disbelief for such words, its seeming irrationality, the subsequent violence that ensues and finally the ineffable wisdom that lay underneath it. Initially the story only seems to be culminating towards a point where Thorfinn eventually exacts his cathartic revenge on Askeladd, his father's murderer. But soon after he dies, things begin to change. Thorfinn is left aimless and his hopes of avenging his father now forever beyond his grasp. Neither having any purpose to live nor a reason to die, Thorfinn finds himself laboring in a farm where he meets Einar whose friendship helps him confront the demons he buried. Demons who take the form of those he had slayed in his quest for vengeance.

Thorfinn is a character who is given every possible justification to be violent. He is an exceptionally skilled warrior and is singularly determined when it comes to combat. He lost his father to a band of mercenaries when he was young and spent most his life filled with resentment. Here is a man who has endured untold suffering in his life and is perhaps rightly justified in regarding those around him as his enemies. And yet how could he not have any enemies?

The cycle of violence that we observe in Vinland Saga has one curious element to it. Everyone has a reason to kill. Some kill for sustenance. Some for glory. Others because they have simply been raised as warriors and cannot envision another manner of living. Violence is shown as being almost a part and parcel of human nature. An inevitable consequence for any person who wants to protect himself or his family. In such a world, renouncing violence would be a sign of brazen stupidity, tantamounting even to death. But Vinland Saga shows us that it might also ultimately be our path to salvation.
I believe what sets apart Vinland Saga is that it doesn't settle for the conventional narrative of good and evil where good eventually triumphs. It cuts through the heart of the problem by weaving a story which not only shows the reasons behind the decadence of human beings but also the strength of their will to break out of it. It seems to me that the statement "You have no enemies" does not represent a factual claim. There are certainly many things out there in the world that seek to hurt and kill, the worst of which as Vinland Saga shows, are members of our own kind. What it does represent instead is an act of faith which stems from a belief in the fundamental virtue of human nature. An act of faith which also emerges from an understanding that regardless of how much we consider ourselves to be fettered by circumstance, we can still break free from causality. Showing what has been and what could be is relatively simple. But to show where what has been becomes what could be is I believe what makes Vinland Saga truly extraordinary.

Ctrl-C Book Club: A People's History of Computing in the United States by *~lettuce* and other participants

This summer on Ctrl-c's iris message board we tried out a first experiment with a "book club" discussion thread (in thread 423). This article is a summary of the Book Club discussion idea and some responses.

The book I proposed we read is A People's History of Computing in the United States, by Joy Lisi Rankin, published by Harvard Press in 2018. The book would be of interest to anyone participating in Tilde computing. I kicked things off in the discussion with some summary and questions posed, and let potential participants in the discussion know that while the book was a great read, it was not necessary to read it to be involved in discussion.

The book in the introduction sets itself up as a counter to the "great man" theory of computing, which I'll summarize as "First there was the military building computers and funding labs. Then there was Bill Gates. Then there was Steve Jobs. Then there was Mark Zuckerberg..." Instead, it tells of the communities of people building their own alternate computing systems.

In Chapter 1, "When Students Taught the Computer", it describes the late 1950s to early 1960s period at Dartmouth when Professor Tom Kurtz had an ambitious idea that all students taking basic Intro Mathematics at the university, which is part of the general required curriculum, should receive basic instruction to working with computers in their class. And all should be given access to computers to use. And all of this without having to submit programs as punch +paper, or sent (on paper) to others, the "elites" who would do the actual physical computing and send the results back days or weeks later after running it for the students. Of course, one problem was lack of computers. So they started fundraising. Then Kurtz learned about timesharing. A single computer could have lots of terminals connected to it. The timesharing computer had software that split up its cycles, allowing multiple programs to run by different users simultaneously all wired in to the same computer. Kurtz decided to build such a system, but one thing was of utmost importance to him...

Kurtz instructed his programmers [of the time-sharing system], 'In all cases where there is a choice between simplicity and efficiency, simplicity is chosen. Every effort will be made to design a system convenient for the user. But maximizing the time that the 235 [computer] is used is NOT one of the goals.' Kemeny and Kurtz demonstrated their commitment to users in their memo outlining the time-sharing system. Rather than starting with an explanation of the computer processes involved, they explained time-sharing from the user's point of view - how a user accessed the machine, the commands entered by the user, and how the user exited the system. Only after that description did they detail the computer processes connecting the teletypewriters, the Datanet-30, and the GE- 225 computer. The prioritization of ease of use extended to BASIC. Kemeny worked to write a language that could be learned in layers: once a student had mastered a beginner's level, with which he could still write powerful programs, he could supplement his skills with additional commands and programming techniques, but nowhere along the way would his advanced techniques detract from ease of use for the beginners. These values of widespread use, user convenience, and a privileged position for BASIC - the values of individualized interactive computing -became embedded in the system." (pages 29-30)

As a starting point for discussion, I posed the prompt: we have a shared timesharing (Linux) computer that we all use and operate together. How or where in our computer/tilde/ctrl-c community do we have an environment "convenient for the user?"

I was also wondering how are we or how could we emulate the scaffolding described above about BASIC? What are the beginner-level introductions that are provided for newbies to learn the most basic skills, such as navigating the file system, accessing email, participating in the iris message board, using IRC chat, creating and editing text files, and anything else that would be helpful to know at the beginning? How do we provide a basic knowledge of these things and then a pathway to learning more? Finally, I posed the question: "Are we embedding our values in this computer system?"

I started the discussion by providing some of my own responses, first to the latter questions. I think in some ways we provide some intro pathways, with the Message of the Day (MOTD), and generally with the

overall activity of the community on iris/IRC/CTRL zine, but we may be missing some opportunity of welcoming folks by leaving out a handbook, introduction experience and instead expecting folks to just find a Linux book or tutorial to get started. I think a lot of learning MORE works really well by creating a space for folks to ask and answer questions here in our beloved iris. But for example, when we've tried to restart a wiki, that didn't jumpstart as quickly. Even the idea of providing an introductory welcome text file or mini intro handbook could be a good way to let beginners know they are welcome and how to participate. Coming back to the text I posted above: I think this is because there may not be an obvious way for us to provide an introduction to beginners on a Linux system, and I was wondering folks' thoughts on that.

Stated differently (or in addition): where does our computer system emphasize simplicity and "convenience for the user" over and above "efficiency" of the system?

Lots of folks participated in the discussion. Rather than reprint the thread directly I'm going to summarize some of the responses.

One of the discussion points was on the Message of the Day providing a reminder of the `newstuff` program that lets you know the most recent webpage, gemini pages and iris messages posted. So many programs or options are hidden in the command line, hard to discover in the Bash shell, so potentially a beginner-shell could have a menu system with some common options such as starting a bash shell, starting iris, starting IRC with an easy client, launching the nano editor, viewing the wiki, or disconnecting.

There was a mention that a mini-handbook could be a nice addition, and that directing new users to iris specifically is a great approach as it acts like the "local greasy spoon diner that's been there forever."

Some of the discussion participants are professors and talked about the distance between that early computing era and today, and a fear that today's students eyes may glaze over when presented with low-level computing or the command line. Some responses to this discussed showing utility before showing the underlying system, and motivating

students' desire to learn by including games (such as the original BASIC games books/programs) or programs like bashcrawl which teach the user basic Linux commands through playing a simple text adventure.

Some other shared ideas included creating a section in Ctrl Zine with little tutorials or tips and tricks for beginners.
Finally in the discussion several folks pointed out an idea from Chapter 1 of the book of computer access as akin to library access. "I'm a big fan of public libraries, and so this really lands with me," said ~nttp. Finally, ~pgadey posed the question "If public access computing is like open stack library access, then what is a membership of a pubnix like ctrl-c.club?"

And that was our first book club discussion on iris. Thanks to the Ctrl-C community for trying out this experiment, and particular thanks to ~kiseratu, ~nttp, ~pgadey, ~loghead for your contributions to the conversation.

^Z