

The Gopher Times

Opus 5 - Gopher news and more - Jun. 2022

Bitreich Con 2022, Come and Talk!

20h

Greetings at 852.770114854 km/h, 34943.004 miles over the Atlantic Ocean.

This is a happy reminder, that in less than 30 days, brcon2022 will happen.

There will be two parts:

July 25th to 28th Online presentations, then one day to get to Belgrade

July 30th to 31st We will be in presence, having fun in Belgrade, Serbia.

If you want to hold a presentation of your interest, please see the Call for Papers: [1] and send your proposal to

`1 gopher://bitreich.org/1/con/2022`

Christoph Lohmann <20h@r-36.net>

There is already a wide variety of topics registered, from medicine to simple software over geology and hopefully a special greeting from our science supervisor Prof. Skildgaard who wants to give advices to all of us humans.

See you online and in presence!

Sincerely yours,

20h Chief Conference Officer (CCO)

Animated ASCII art

linuxconsole

With all the history of ASCII art and demoscene, it would be a shame if noone ever tried to combine the two in animated ASCII art. Courtesy of textfiles.com, we can browse through a collection of 93 animated ASCII pieces of arts. [1]

They are also mirrored at the bitreich gopher site [2]

The animation speed will likely be too high for a terminal, and can be slowed down with the throttle(1) program as advised by linuxconsole.net, or with pv(1) as below:

```
1 http://artscene.textfiles.com/vt100/  
  http://linuxconsole.net/ascii_art.html  
2 gopher://bitreich.org/1/vt100/animations/
```

```
curl -s gopher://bitreich.org/1/vt100/animations/twilight.vt | pv -qL3000
```

You may use the "reset" command to get your terminal normal again after watching.

Some are just a pun, a few frames to only give impression of movement, while other might be closer to a short animated movie. Talking of which, long movies were also done:

```
https://www.asciimation.co.nz/  
telnet towel.blinkenlights.nl
```

These characters transmitted through one protocol or another, whispers to us, through our terminal screen, tales from the imagination of plain text artists.

Prof. Skildgaard: Only Turtle Fans

20h

I am happy to announce, that the scientific head of bitreich, Prof. Skildgaard, the professor for slow sciences at the Aarhus university in Denmark, now has opened his own website [1]

You can see many #turtlefan pictures. [2]

Please recommend his work! He has done so much for us, like reviewing all entries to the last and the coming bron. This takes ages!

Sincerely yours,

20h Chief Slowness Executive (CSE)

```
1 http://onlyturtlefans.com/
```

```
2 <anna> #turtlefan: gopher://bitreich.org/1/memecache/turtlefan.png
```

Synthetic ASCII Art

tgtimes

When an entirely new way to solve problems is discovered, all sorts of medias, and not only the tech-oriented ones, are fond to publish abundantly about it. Be it quantum computing, blockchains, machine

learning... Shortly after a new big toys like these comes-up, hackers come, and start experimenting with it, sometimes coming-up with entirely new way to use it.

This time we are reviewing the combo of Machine Learning and ASCII art.

I was expecting to present cute attempts at drawing images with computer-made text, but this is nothing of the sort. Be prepared to see Science at the service of Art.

Generated Typewriter Art This research paper (no less!) shows that it is possible to write software for placing characters, later typed during 6 hours by a human operator (for this example). It is unsettling to see details much smaller than the characters themselves be drawn on paper, along with shades of grey of various intensities. [1]

Generated ASCII Art in 2010 This is possibly the state of the art of 2010 technology. It was announced in the yearly conference SIGGRAPH hence presented to an audience full of computer graphics engineers. The work of three researchers from Hong Kong, Xuemiao Xu, Linling Zhang and Tien-Tsin Wong, shows results of surprising accuracy. The story does not tell whether there ever was a job offer "looking for ASCII artists for a scientific experiment" posted on the job board of the Chinese University of Hong Kong. While the paper contains the complete math used, it also illustrates and explains methods to achieve this level of accuracy. And no, it is not exactly machine learning, but hand-crafted strategies, combined statistics and other data massaging. After all, it was published five years before things like Tensor

Flow were introduced... [2]

Generated ASCII Art in 2017 Is seven years enough time to improve upon that previous achievement? Quoting the previous paper as well as others in its own work, Osamu Akiyama of the Osaka Faculty of Medicine kept the ball rolling. This throws the big guns of machine learning to reach higher skies. Its input data were Japanese BBS such as 5chan (2chan) or Shitaraba, which extends the ASCII set to all of unicode, notably the CJK set. If the result of the paper are not enough to convince you, the "Bad Apple" often used as a video demo in the Asian market have been converted in its entirety. Something out of reach if doing every frame by hand. The TensorFlow and Python code used is released publicly, and an online demo is offered for the curious. [3] [4] [5] [6] [7]

Is it so futile? Not so sure. After all, representing anything with a computer is a matter of making a reality fit onto something terribly awkward and unnatural: a display. The pixels, the square elements praised for providing a grid to throw data at, are promising, but themselves have their quirks to be worked around. For instance, sub-pixel geometry uses the same techniques as those presented by these papers for improving the realism of images beyond what a single pixel can offer. It is, for ASCII art like for anything else, a matter of representing something, real or fictitious, through a medium of some kind.

ASCII art has the ability to fit an image somewhere where there could only be text. For the example of a train station concourse with a large split-flap display: for displaying a big arrow at the end of the service, re-

placing the display by an equally large color screen can be costly and much more power-hungry, while an ASCII arrow on that existing display would be consuming no power for that still image.

1 <https://graphicsinterface.org/wp-content/uploads/gi2021-13.pdf>

2 <http://www.cse.cuhk.edu.hk/~ttwong/papers/asciiart/asciiart.html>

3 https://nips2017creativity.github.io/doc/ASCII_Art_Synthesis.pdf

4 <https://nips2017creativity.github.io/>

5 <https://yewtu.be/watch?v=8GulN69Cgbg>

6 <https://www.vice.com/en/article/zmymwx/machine-learning-ascii-art-neural-net>

7 <https://github.com/OsciiArt/DeepAA>

BIG BROWSER IS WATCHING YOU!

20h

Are you feeling watched all the time? Do you feel unsure when doing something nasty? It is true, you are watched: By BIG BROWSER. Whenever you use the web, someone else is masturbating to your web history.

You want to know how to be able to do nasty things online without someone masturbating to it? Come to br-

con2022 and find out more. [1]

This time online and in presence!

See you there!

Sincerely yours,

20h Chief Espionage Officer (CEO)

1 [gopher://bitreich.org/1/con/2022](https://bitreich.org/1/con/2022)

Sailing With Grace

tgtimes

The sea! Water all around, not a single piece of land around to stand in, only a single boat that becomes one with you, its captain. Infinite waves under the blue or cloudy sky is all you see for a long trip of many days. Feeling lost, but at the same time united with surrounding nature. After all, the largest part of Earth is covered by the sea.

This is the world of Sailing that awaits each of us, for a single trip hosted by a well proven crew, or as a lone sailor braving tempests after tempests.

Sailing blogs are definitely a good opportunity to dream, the instant of an article.

This blog, *Sailing With Grace*, has taken the decision of offering all its content through HTTP, but also proxied over Gopher. [1] This recalls an interesting point: it proves that Gopher is not only good for talking about Gopher and computer things, but is also oriented toward the out-

1 gopher://gopher.sailingwithgrace.com

side. Is it ready to be used by people who are not gopher geeks?

It always was to begin with, so why would it not? Are people less able to use computers now than they was before the web came? The discussion is open.

sfeed 1.5 Released

Hiltjo

sfeed [1] is a tool to convert RSS or Atom feeds from XML to a TAB-separated file.

sfeed has the following notable changes compared to 1.4:

- `sfeed_curses`: interrupt `waitpid` while interactive child program is running. This now handles `SIGTERM` on `sfeed_curses` while an interactive child program is running.
- `sfeed_curses`: close `stdin` before spawning a plumb program in non-interactive mode, which is more intuitive: the program doesn't

1 git://git.codemadness.org/sfeed
gopher://codemadness.org/1/git/sfeed
<https://codemadness.org/releases/sfeed/>
gopher://codemadness.org/1/releases/sfeed/

seem to hang when it expects input in this case since there is no way to send input anyway.

- Properly escape backslashes in the man pages (thanks adc!).
- Documentation improvements to the man pages and a progress indicator example script for `sfeed_update`.

I want to thank all people who gave feedback.

Thanks, Hiltjo.

Wireless, wireless everywhere

tgtimes

Wires! Cables! Connectors! Computer and electric systems seems to befriend with plugs and sockets. Why is the computer industry running away from them for everything exposed to users?

Where do I plug the cable? Everyone needfully face this question at least once, be it the first time they own a computer. From the various connector shapes to choose from, to the various set of protocol the Universal USB connector supports,

cables provoke confusion to cable-haters and computer neophytes.

Cables are ugly It might not be true for everyone, but computer manufacturers seems to say differently. Starting with the name "wireless", that comes by opposition to wires, supposing they were something to avoid. Cable management is a full time job for datacenter jockeys, and a chore for the cable-hating computer user.

Cables are immobile Unless making use of an uncommon cable management strategy, objects connected to cables cannot be carried too far away without unplugging everything devices are connected to.

So here comes wireless. While not frequent in large computer infrastructure, wireless is invading the market along with battery devices. Using radio waves to make device talk to each other, at various frequencies, modulation, data rate and distance. Ready to sacrifice any amount of good engineering to make itself more seducing to the market, marketing perpetuates the same illusion of making computer troubles fade away with wireless.

From the Bluetooth protocol swamp of mixed edge-cases and complexity, to the security vulnerabilities of Wi-Fi, to the security vulnerabilities of Bluetooth, to the proprietary but popular protocols like LoRaWan, to the unreliability and instability as opposed to wires, to the black box of wireless broadband such as UTMS and LTE, Wireless does not have the same fame among developers valuing simplicity and reliability.

Even the United Army holds griefs against wireless such as Bluetooth, and disrecommand it for use by militaries: [1]

>> Do not use Bluetooth devices to send, receive, store, or process classified information.

This means no Bluetooth keyboard, no Bluetooth headset during phone calls, no Bluetooth sharing between the phone and the computer... In other words, no Bluetooth.

Nonetheless, wireless is fun, beautiful, and filled with culture. While marketing pushed engineers from the wireless cliff, long before computer came, radio waves were put at good use in the most simple forms: radio communication. From the AM and FM radio stations to listen while on the road, the medium-range boat, airplane, truck, pedestrian talkies, and even satellite communications, hobbyists building-up their own antennas for inter-continental communication, garage door openners and remotely controlled drones...

Complex and twisted wireless protocols are only a special case of radio communication, and simple unobfuscated methods of communication are possible, and even frequent.

Be it a simple and inexpensive RTL SDR dongle receiver [2] or a complete receiver-emitter such as HackRF [3] or LimeSDR, [4] many gears exist for experimenting with radio transmissions.

Every year, the American Relay Radio League (ARRL) is publishing a large book focused on radiocommuni-

cation, and its chapter 1 section 1 is *Do-It-Yourself Wireless*.

This is an invitation for everyone to discover or rediscover the universe of electromagnetic fields communication.

1 https://armypubs.army.mil/epubs/DR_pubs/DR_a/pdf/web/ARN4771_Pam25-2-9_Final_Web.pdf

2 <https://www.rtl-sdr.com/>

3 <https://greatscottgadgets.com/hackrf/one/>

4 <https://limemicro.com/products/boards/limesdr/>

Open-Source Breathing

tgtimes

The previous opus had a word or two about how difficult it could be to get open hardware medical devices. The Freespireco [1] project aims to bring a respirator device to life as a completely Open Hardware project.

The challenge is not coming-up with something that works and is reliable, but instead to provide a structure robust enough to be accepted (and funded) for performing all the necessary certifications needed before being allowed to the medical device market.

There are usually categories of criticalities, and an artificial respirator is not escaping to the rule. The organiser of the project have pursued this goal since long, and might likely have a very long road to go.

It is essentially a pioneer of Open Hardware for critical medical devices, step-by-step paving up the road toward certification: designing and building devices to test these equipment, issuing standards for data (like a JSON schema received over a serial

port directly from the device).

The strategy: offering reproducible tests as an anchor for trust. Precious argument for facing big pharma equipment vendors that are having an interest in locking their device down, preventing repair or even inspection.

In a same journey toward braving Goliath: accessing the Outer Space. And it is, as crazy as it looks, far from impossible to contribute to space research even without a diploma: The RTEMS [2] project is open to contribution.

But that does not discourage the authors of the respirator project to keep going. Not in a blind trust for the medical industry, but in full foresight that nobody would want its mom's life given to a hobbyist toy made in a garage. With this reality in mind, "whatever it takes" turns into "whatever is done", and the road to certification progresses, one breath at a time.

1 <https://www.pubinv.org/project/freespireco/>

2 <https://rtems.org/>

This project existed since a while, and kept improving. In this interview with 20h, he shows us what Geomyidae's got under the hood.

>> What is Geomyidae?

Geomyidae is a Unix/Linux daemon/service serving the gopher protocol.

>> So what is gopher?

Gopher here is an internet protocol, which was first developed at the University of Minnesota. After its short

success, it declined, but is now striving again to be used for its simplicity and hierarchy. For better visual display of your gopher experience, use something like links, lynx or sacc. Those are gopher clients.

>> How does Geomyidae help with getting started with gopher?

The installation of Geomyidae is very simple. You can setup your Geomyidae right away:

```
git clone git://bitreich.org/geomyidae
cd geomyidae
make
curl -s gopher://localhost:7070
```

Yes, curl supports gopher! And it supports gopher and TLS too!

>> Are there many alternatives among gopher daemons?

Yes, there are many. Some are there due to historical reasons, others have gone out of shape over time. One of the most popular alternatives is pygopherd.

>> How does Geomyidae compares to other implementations?

Geomyidae offers a unique simple way of expressing gopher content. See the manpage or the examples in the source for how .gph files are formatted. And it does just what you want it to do. No strange abstraction files like in the original gopher daemons are the default way. In the

newest release of Geomyidae compatibility scripts were added. But those are to preserve the unique history of gopher.

>> Did Geomyidae have significant evolutions since the beginning?

Yes. Originally Geomyidae was named gopherd for Plan 9. It then was ported over to Linux. On Linux it was renamed to Geomyidae. During that development much has happened: There were significant speed-ups (due to the patches and work of other people!), features were added especially in new dynamic content handling. You can easily see all features in the documentation and especially the simple manpage.

>> Does Geomyidae work with all gopher clients?

Yes. Geomyidae supports the original protocol from the beginning, up to modern gopher with TLS. For the intermediary gopher+ protocol there is a compatibility layer.

>> Has NSA inserted a backdoor onto Geomyidae?

I am not allowed to tell you.

>> How does gopher help with privacy?

The gopher protocol has the unique property that all data you send over the line can be easily controlled and seen. This is different to HTTP, where headers, HTML and Javascript got so complex, it is uncontrollable. See the gopher onion project [1] for how to combine this with tor for total privacy and anonymity.

>> Are there TLS support on some gopher clients already?

There is support in curl, mpv/ffmpeg, sacc and more. It is very easy to add TLS support to any client. You simply connect via TLS on the gopher TCP port (default: 70) and if it works, keep that connection open.

>> Are there been any evolution of the gopher protocol itself since the beginning of Geomyidae?

The technology used is simple. Gopher does not allow requests, which begin with the first bytes of a TLS request. So any proper and old gopher daemon will simply refuse the connection. Then the client is free to reconnect without TLS based on its security configuration. Any ISDN line

will handle such probing requests for TLS easily.

>> Did Geomyidae have to adapt itself to the gopher protocol? Did it make gopher change?

Geomyidae changed the part of gopherspace it was able to reach. Many servers run on Geomyidae. There is software written just for Geomyidae and its gph format. The TLS extension of the protocol came from Bitreich / Geomyidae. We also set the standard to simply use UTF-8 as default representation in gopher menus and so bring it into the 21st century. I can conclude: Yes, Geomyidae changed and will change gopher.

>> Have you seen Geomyidae ever used outside a hobby project?

Well, Bitreich is serious in changing the software world. Most of gopherspace is »hobby projects«. But then, most of gopherspace is made from heart blood and love, which makes it part of the life of the authors.

>> Is Geomyidae ready for non-hobby uses?

Geomyidae is ready for any use. It is stable and optimized to scale better than the cloud.

>> Geomyidae uses ".gph" files.

Does it replace the gophermap standard? Yes, in Geomyidae it does. Gph is simpler and easier to adapt to, especially when you come from some markup world.

>> Does Geomyidae support dynamic pages?

Geomyidae supports two forms of dynamic pages: One which uses the gph markup and one, where the script/application outputs raw gopher output. Additionally it supports in the latest release a form of REST, where paths are transformed into arguments to scripts. There is also support for `index.dcgi/index.cgi` scripts to have better looking paths and URIs.

>> Is Geomyidae already packaged in some Linux/BSD distributions?

As far as I know it is packaged in gentoo, Archlinux (and more), all BSDs. Since it is so simple to package: Just extract the tarball, run `make` and `make install`, the packages are easily made for any package manager.

>> What is planned for the next releases of Geomyidae?

As of now I have worked through my whole long-standing TODO list for Geomyidae. New ideas will evolve from people sending in patches or

```
1 gopher://bitreich.org/1/onion
2 gopher://bitreich.org/0/documents/bitreich-manifesto.md
3 ircs://irc.bitreich.org/#bitreich-en
```

through practical need. Geomyidae follows the Bitreich manifesto [2] where a software can be done.

>> How to get involved? Getting help, discussing, bug hunting, code contribution, documentation...

If anyone wants to get involved, first download Geomyidae, run it, have fun using it, creating gopher content. If you run into problems, have patches or suggestions, come on IRC [3] and discuss with us your problem. For e-mail, send such requests to `20h@r-36.net`. All contact is in the manpage too.

>> Can I have an ice cream?

Yes, you will get one, once we meet again.

Embedded Forth Programming

tgtimes

Big computers can run large and complex programming languages, so what can small computer run?

Compiled languages, in particular those with a small runtime are often chosen. But the interpreted languages also have an audience willing to code with their favorite programming environment for them. Programming lan-

guages as big as Python have their embedded counterpart (MicroPython) thanks to significant efforts. They serve their purpose to embedded enthusiasts as educational and scripting languages to many.

But small "language in a nutshell" are fitting right the small resources of microcontrollers. This is the case of

Forth and its stack-machine approach.

Mecrisp This implementation immediately targets microcontrollers. See for instance the work of librehacker.com author Christopher Howard. [1]

chipFORTH Another implementation of Forth, which were used by NASA [2] for improving reliability of its flight control system, among the mosts critical pieces of software of a shuttle.

<https://github.com/corecode/forth> Among notable Forth projects is Simon "corecode" Schubert's nimble forth implementation as well as hardware code describing the working of a CPU that executes Forth natively [3]

<https://forth.chat/> If feeling like having a taste of Forth and Forth community, there are several channels featuring forth that you could enjoy, some of which are oriented toward hardware projects directly [4]

<https://github.com/chmykh/apl-life> This is Conway Game of Life in APL in Forth What a long chain! It is APL programming language implemented in Forth, and Conway game of life implemented in APL

<https://github.com/remko/waforth> Feeling like pushing the irony of "Web" assembly even further? Why not blasting a Forth implementation at it? [5] This proves Forth as the new programming language *en vogue*

<http://collapseos.org/> What else does a programming language need to prove itself useful? A kernel? Check! Collapse OS is an operating system targetting resilience beyond extreme, as it is designed to resist everything around it tearing apart, including the whole civilisation. When nothing remains but wastelands, CollapseOS will be there for a rebirth of civilisation out of computers made from scavenged parts. Civilisation is rising and falling all of the time, just not all parts at the same time.

>> Forth is, to my knowledge, the most compact language allowing high level constructs. -- Collapse OS author.

[gopher://retroforth.org/](http://retroforth.org/) <https://retroforth.org/> A forth implemented in C, Python, C#, Nim, JavaScript and Pascal! The C version permits to embed the script into a binary along with the interpreter, for a single-binary deployment process. The more classic way to use it is to use shebangs scripts to have executable scripts.

Many smaller utilities can already provide something you needed:

<http://retroforth.org/examples/Casket-HTTP.retro.html> An HTTP server

<http://retroforth.org/examples/Atua-WWW.retro.html> A Gopher to HTTP+HTML Proxy on top of Atua.

<http://retroforth.org/examples/Atua.retro.html> A gopher server, already listed on the Gopher index of links, the Gopher Lawn [6]

<http://retroforth.org/examples/7080.retro.html> A s

<https://gitlab.com/goblinrieur/spreadsheet/> A spreadsheet application in the terminal.

gopher://forth.works:100 This is a collection of code blocks written in the Retro Forth's author (crc) newest Forth implementation. It is itself served by a gopher server (blocks 203-205 on the list above) in Forth.

<https://github.com/oriontransfer/pl0-language-tools> A PL/0 implementation in Python that can emit Retro Forth code as output. It looks like Forth simplicity, portability, stability and speed of execution made it a good candidate as a target language. The PL/0 language is known for the book *Algorithms + Data Structures = Programs* from Niklaus Wirth, himself famous for the Wirth Law:

>> The hope is that the progress in hardware will cure all software ills. However, a critical observer may observe that software manages to outgrow hardware in size and sluggishness. -- https://en.wikipedia.org/wiki/Wirth's_law

<https://ribccs.com/candy/> If you were doubting about Forth being fit for the industry, bear in mind that the above is a very-large scale VFX Forth project with over a million lines of code!

<http://sam-falvo.github.io/kestrel/2016/03/29/vibe-2.2> Why not spin a vi-like text editor itself in forth? See how few code it takes to implement one.

<https://git.sr.ht/~vertigo/shoehorn> An answer to the bootstrapping problem: how to get from no software to a complete system? Which compiler compiles the first compiler? Forth's simplicity is a good candidate for solving this problem.

<https://git.sr.ht/~vertigo/forthbox> Software environment for computers to base upon right after booting: a system shell in forth with real hardware projects dedicated to it. Think of a LISP machine, but instead being a Forth machine.

<http://deathroadtcanada.com/> This video-game uses Forth as a scripting language. When a whole scripting language fits on a thumb, putting it everywhere costs nothing!

Such a large tool chest for such a small language. With the Covid, Wars under disguise, and other supply chain troubles, the demand of fea-

ture stability rises face to the traditional "more features". In these trying times, anyone is welcome to go Forth.

```
1 gemini://gem.librehacker.com/gemlog/tech/20220331-0.gmi
  gemini://gem.librehacker.com/gemlog/tech/20220305-0.gmi
2 https://www.forth.com/space-shuttle-instrumentation-interface/
3 https://github.com/corecode/forth-cpu
4 ircs://irc.hackint.org/#forth-hardware-projects
5 https://el-tramo.be/waforth/
  https://el-tramo.be/thurtle/
6 bitreich.org/1/lawn/c/gopher.gph
```

A new IRC network: IRCNow!

tgtimes

A new IRC network is in town! [1] Ever wanted to feel what an early community looks like? The admin jrmu brought the project together, and is currently collecting users along the way.

Whether you looked for a place to host your own community, or wanted to see a fresh community grow from fertile ground, the community is welcoming and active.

>> IRCNow: Of the Users, By the Users, For the Users

Something else from this community might catch your attention, is its orientation toward being administrated by its users themselves: rather than letting the founder handle everything, the community is oriented toward serious teaching of unix command line and system administration to anyone, from beginners to advanced users seeking improvement.

In-person teaching sessions were covered during the LibrePlanet 2022 event [2] with recording of a test-run of the event [3] where future and present hackers met together working on their system administration and community building skills. Linux Magazine also ran an interview giving a good impression about the spirit of the project: [4]

Beyond yet another IRC network to chat with, IRCnow offers hosting services for IRC bouncers, Bots, E-Mail, VPN, Code, File Storage, and Shell Accounts.

The wiki itself features plenty of technical information on system administration as a support for its bootcamps, which offers a comfortable step-by-step introduction to a complete server administration. [5] I have seen system administrators getting hired knowing less than this!

```
1 irc://irc.ircnow.net:6667
  ircs://irc.ircnow.net:6697
2 https://jrmu.host.ircnow.org/libreplanet/libreplanet.pdf
3 https://0x0.st/oTal.webm - 0h20m: audio starts - 1h15m: talking about Gopher
4 https://www.linux-magazine.com/Issues/2021/249/Interview-IRCNow
5 https://wiki.ircnow.org/index.php?n=Minutemin.Bootcamp
```

Do you happen to be a podcast enjoyer? Maybe you considered to have something to listen to on the road or while cooking.

Combining many different sources, you may encounter some heirlooms by searching through this gopher front-end for podcast search. [1]

The platform aggregates multiple search APIs of RSS link aggregators with a focus on audio podcasts, and extracts the RSS links for you, so you do not have to search through a dozen of webpages just to find the RSS button.

For instance, knowing about the Amp Hour podcast, I tried searching for it: "Amp Hour" in the search field, and

```
1 gopher://gopher.icu/1/pod
2 ircs://irc.bitreich.org/#bitreich-en
```

bingo! The first result is "The Amp Hour Electronics Podcast", that was quickly added to my list of RSS feeds in a blast.

Being based off Gopher, this makes it insanely easy to automate a script searching for podcasts, then downloading the entries and uploading them to an MP3 player of any kind (dedicated, or as part of a phone or other portable computer).

Want to know more about it? One place to discuss about it is the Bitreich IRC server [2]

Relics of Fast Fourier Transform

rue_mohr

In 1967, the Kooley-Tukey FFT algorithm (the one we all use now) was written in Fortran. What the hell

were they running it on, and what damned data were they feeding into it?!

```
SUBROUTINE FOUR1 (DATA, NN, ISIGN)
C THE COOLEY-TUKEY FAST FOURIER TRANSFORM IN USASI BASIC FORTRAN
C TRANSFORM(J) = SUM(DATA(I)+W**((I-1)*(J-1))). WHERE I AND J RUN
C FROM 1 TO NN AND W = EXP(ISIGN*2*PI+SQRT(-1)/NN). DATA IS ONE-
C DIMENSIONAL COMPLEX ARRAY (I.E.: THE REAL AND IMAGINARY PARTS OF
C THE DATA ARE LOCATE IMMEDIATELY ADJACENT IN STORAGE, SUCH AS
C FORTRAN IV PLACES THEM) WHOSE LENGTH NN IS A POWER OF TWO. ISIGN
C IS +1 OR -1, GIVING THE SIGN OF THE TRANSFORM, TRANSFORM VALUES
C ARE RETURNED IN ARRAY DATA, REPLACING THE INPUT DATA. THE TIME IS
C PROPORTIONAL TO N*LOG2(N), RATHER THAN THE USUAL N**2. WRITTEN BY
C NORMAN BRENNER, JUNE 1967, THIS IS THE SHORTEST VERSION
C OF FFT KNOWN TO THE AUTHOR, AND IS INTENDED MAINLY FOR
C DEMONSTRATION. PROGRAMS FOUR2 AND FOURT ARE AVAILABLE THAT RUN
```

```

C      TWICE AS FAST AND OPERATE ON MULTIDIMENSIONAL ARRAYS WHOSE
C      DIMENSIONS ARE NOT RESTRICTED TO POWERS OR TWO. (LOOKING UP SINES
C      AND COSINES IN A TABLE WILL CUT RUNNING TIME OF FOUR1 BY A THIRD.)
C      SEE-- IEEE AUDIO TRANSACTIONS (JUNE 1967), SPECIAL ISSUE ON FFT.
      DIMENSION DATA(1)
      N=2*NN
      J=1
      DO 5 I=1,N,2
      IF (I-J) 1,2,2
1     TEMPR=DATA(J)
      TEMPI=DATA(J+1)
      DATA(J)=DATA(I)
      DATA(J+1)=DATA(I+1)
      DATA(I)=TEMPR
      DATA(I+1)=TEMPI
2     M=N/2
3     IF (J-M) 5,5,4
4     J=J-M
      M=M/2
      IF (M-2) 5,3,3
5     J=J+M
      MMAX=2
6     IF (MMAX-N) 7,9,9
7     ISTEP=2*MMAX
      DO 8 M=1,MMAX,2
      THETA=3.1415926535*FLOAT(ISIGN*(M-1))/FLOAT(MMAX)
      WR=COS(THETA)
      WI=SIN(THETA)
      DO 8 I=M,N,ISTEP
      J=I+MMAX
      TEMPR=WR*DATA(J)-WI*DATA(J+1)
      TEMPI=WR*DATA(J+1)+WI*DATA(J)
      DATA(J)=DATA(I)-TEMPR
      DATA(J+1)=DATA(I+1)-TEMPI
      DATA(I)=DATA(I)+TEMPR
8     DATA(I+1)=DATA(I+1)+TEMPI
      MMAX=ISTEP
      GO TO 6
9     RETURN
      END

```

And no, you **cannot** get the IEEE document because IEEE broke it up into pages and sells each page individually.

```

"PROGRAMS FOUR2 AND FOURT ARE AVAILABLE THAT RUN
C      TWICE AS FAST AND OPERATE ON MULTIDIMENSIONAL ARRAYS WHOSE
C      DIMENSIONS ARE NOT RESTRICTED TO POWERS OR TWO."

```

But, this code was easy to port because it was small, so, to this day, we use it. It was ported from Fortran to BASIC, then to C, then to C++ and everything else.

Nobody ever actually understood it, so they didn't fix anything. You see, Fortran has no bitwise operators, so

alot of the acrobatics in that code are just doing bitwise operations in regular math. Its absolutely amazing when you tear it apart.

I got the code from a bad scan of a document off a military ftp site. What I love, and find halarious, is that this code has been ported and

hacked a million times since it was written.

But, from the comments, it, itself, is a hack. It is a mash up of cooley and tukeys code. It is a hack, from 1967.

Maemo Leste keeps kicking in!

tgtimes

The ultimate hacker's toy project: a OpenSource powered hand-held computer.

Where to start from? There can be two walls preventing every Linux enthusiast from having its own phone with a "Linux Powered" sticker on it:

1. hardware support: getting Linux to boot on the twisted hardware setups of smartphones can be frustrating.
2. application support: writing all the tools that make a plain unix shell useable as a phone, that we usually take for granted on a phone operating system. It may be as simple as a daemon watching incoming phone call from hardware abstractions (those from in 1.) and playing a ringtone.wav whenever a call comes in, it still has to be written. Same goes for a keyboard application if it uses a touch-screen. Same goes for anything.

Since it goes beyond the scope of a week-end hack, collaboration takes place for making these projects happen.

Maemo Leste is now existing since more than four years, and keeps being developed at good pace. It even shines where Android does not: it uses mainline Linux kernel instead of forks that never get upgraded nor contributed back to Linux. This means that all software officially supported by Maemo Leste might also be available to many more Linux-based projects.

Of course, there are non-official porting efforts for more hardware under-way to become a completely supported target. Like it is for every operating system project.

Maemo Leste, the project bringing a real UNIX shell where you only had a Android Java ecosystem, featuring GPS chips reverse engineering, and a working phone module.

The support for the inexpensive Pine-Phone means you can get a fully working linux phone in your pocket. Grab it while it is hot, the lack of bloated prebuilt application forced into it by the vendor means it will not catch fire! [1]

¹ <https://maemo-leste.github.io/maemo-leste-sixteenth-update-november-and-december-2021-january-april-2022.html>

The post-Snowden era is marked by a new fact that cannot be ignored anymore: NSA (among others) is watching you (among others).

Does that change anything to my everyday life? Probably not, they already were before you knew about it. Should I do anything about it? No answer. The eternal doubt that modern society is famous for:

>> I do not know, do not ask me. That question is weird anyway. Let me go back to my life.

That same doubt that occurs when you look up on a supermarket and see the mess of wires, tubes, cables and neon lighting, barely even hidden, at best painted in white... The worst scene of industrial warehouse, as if taken straight out of the Brazil [1] movie.

A landscape that is in such opposition with the images printed onto every food product being sold, picturing what more or less fits the collective imagery of "house of my grandparents in back-country", promising a natural environment and suggest quality, authenticity, tradition to the buyer... Pictures of a caring lady baking something appetizing, a honest farmer offering a handful of home-grown vegetables or meat... Where did they even find all these landscapes of backcountry without phone line everywhere, tracktors, asphalt, cattle warehouses, wind turbines to put on these product background im-

ages?

>> I do not know, do not ask me. That question is weird anyway. Let me go back to my life.

How did such a landscape, neon dystopia pictures that seems straight out of a /r/cyberpunk [2] post or the latest Blade Runner, got invited into the cozy bubble of the average citizen doing shopping? [3] Who made these places so ugly? Why do I feel like human is being considered like cattle in these kind of places?

>> I do not know, do not ask me. That question is weird anyway. Let me go back to my life.

What weird things am I even saying! It is not like an NSA agent is sitting on every metal beams of these supermarket looking at passersby with an empty gaze. There are cameras though. What do they film? Thieves? Who is checking? Software? Peoples? Are marketing managers looking at these pictures? Of me too? Right now? What do they think of me? Did they look at my hand hesitating between these two products?

>> I do not know, do not ask me. That question is weird anyway. Let me go back to my life.

Going out, one might encounter someone sitting on its empty backpack, with a small cup filled with coins, looking a bit panicked, looking a bit dirty, looking a bit lost, some-

times even a bit drunk, or is it dizziness from living outside? Occasionally they will ask you for another coin to add to their small collection. Passerbys offer them a lie such as "I do not have cash", or a kind word like "no, sorry", keep walking faster without looking, and eventually stops paying the tax and quickly keep going before they got asked for more. What did happen to them? Did they choose to live here? How can I know it will never happen to me? Why do I feel bad if I do not give them what they ask? Why do I feel bad if I give them what they ask?

>> I do not know, do not ask me. That question is weird anyway. Let me go back to my life.

Let's not get fooled or reverse the roles here: Writing this, I am not asking these questions to you, neither you are asking these questions to yourself. The *places* we live in are suggesting these questions.

By building a supermarket out of a warehouse but displaying eye-catching pictures of a scenery that does not even exist, it is obvious that people will notice the disbalance between the two.

By placing cameras filming every square meter of such a place, or even a whole city, it is obvious that people will wonder at some point, who is be-

hind the screen reviewing these images.

The questions are left open. Nothing is made to even give hint about the answer. We are left in the doubt, letting some comfort themself with "it is just in case of a burglary, only a police officer is going to watch" or other claim "they are using these images to study how we think to better control us!"; claims based upon convictions, not facts.

The technician installing these cameras up there has no hint either, its manager just followed the recommendations of the mothership company, itself getting directions from the investor group who purchased the brand, who themself are only trying to keep-up with the trends in that domain.

Why would I care? I stopped to care about these silly questions since long. I came back to the real world for the better. I live my life ignoring what happens around me and it works plenty well.

>> So why is that, at deep down, in the middle of my gut, there is a voice whispering to me that something's wrong. [4]

The thing with living like an ant in the anthill is: you do not get too many answers about how the whole anthill works.

1 <https://www.imdb.com/title/tt0088846/>

2 <https://teddit.net/r/cyberpunk>

3 <https://theuws.com/wp-content/uploads/2011/12/supermarkt.jpg>

4 <https://yewtu.be/watch?v=QcSlAihVM0Q>

IRC is part of the protocols that survived to the advent of the Web.

It still has users, it still has new network and communities initiatives springing out, it is alive.

One single little touch it lacks is end-to-end encryption. Without it, it is perfect for public communities such as software projects discussions and support chat, live event chats... but private 1-to-1 communication could suddenly become a good candidate for end-to-end encryption.

Part of the DarkFi project, this is what Mallumo [1] brings in a simple piece of code using libNaCl, the cryp-

¹ <https://github.com/darkrenaissance/mallumo>

to library from Dan Bernstein, author of ED25519 (in its repackaged lib-sodium form). This is state-of-the-art, well-proven and fast cryptography for end-to-end communication.

With this plug-in dropped in the plug-in folder, all private communication start by a simple key exchange over normal IRC, and the conversation upgrades to nacl-encrypted messages over regular IRC.

There might not be any simpler way to encrypt peer-to-peer communication online.

Publishing in The Gopher Times

you

Want your article published? Want to announce something to the Gopher world?

Directly related to Gopher or not, reach us on IRC with an article in any format, we will handle the rest.

`ircs://irc.bitreich.org/#bitreich-en`
`gopher://bitreich.org/1/tgtimes/`
`git://bitreich.org/tgtimes/`

Did you notice the new layout? We now can jump between single and double column as it is more fit: Some large code chunks will not fit in a two-column layout, but text is more pleasant to read on two columns.