

GNU Planet - Latest News

- [autoconf @ Savannah: Autoconf 2.73 released](#) (2026/03/20 20:00)
Autoconf 2.72 has been released, see the release announcement: <https://lists.gnu.org/archive/html/autoconf/202603/msg00000.html>
- [libredwg @ Savannah: libredwg-0.13.4 released](#) (2026/03/19 06:32)
A major bugfix release. Complete rewrite of the decompressor to fix hairy section reading bugs in some big files. Fixed many dxf roundtrips. See <https://www.gnu.org/software/libredwg/> and <https://github.com/libredwg/libredwg/blob/0.13.4/NEWS> Here are the compressed sources: <http://ftp.gnu.org/pub/gnu/libredwg-0.13.4.tar.gz> (21MB) <http://ftp.gnu.org/pub/gnu/libredwg-0.13.4.tar.xz> (11MB) Here are the GPG detached signatures[*]: <http://ftp.gnu.org/pub/gnu/libredwg-0.13.4.tar.gz.sig> <http://ftp.gnu.org/pub/gnu/libredwg-0.13.4.tar.xz.sig> Use a mirror for higher download bandwidth: <https://www.gnu.org/order/ftp.html> Here are more binaries: <https://github.com/libredwg/libredwg/releases/tag/0.13.4> Here are the SHA256 checksums:
cacff5510f46723462e854e15ecfa97cbc7475acb3eb7ae1ca6e4193ecc2267d libredwg-0.13.4.tar.gz
7e153ea4dac4cbf3dc9c50b9ef7a5604e09cdd4c5520bcf8017877bbe1422cd5 libredwg-0.13.4.tar.xz
cb46bce034296e91cb1a982cd53ec1928b11f4f7f70512dd21513a27959688b5 libredwg-0.13.4-win64.zip Please ignore the broken Source code (tar.gz, .zip) artefacts. They cannot be deleted. [*] Use a .sig file to verify that the corresponding file (without the .sig suffix) is intact. First, be sure to download both the .sig file and the corresponding tarball. Then, run a command like this: `gpg --verify libredwg-0.13.4.tar.gz.sig` If that command fails because you don't have the required public key, then run this command to import it: `gpg --recv-keys B4F63339E65D6414` and rerun the `gpg --verify` command.
- [GNUnet News: GNUnet 0.27.0](#) (2026/03/18 23:00)
GNUnet 0.27.0 released We are pleased to announce the release of GNUnet 0.27.0. GNUnet is an alternative network stack for building secure, decentralized and privacy-preserving distributed applications. Our goal is to replace the old insecure Internet protocol stack. Starting from an application for secure publication of files, it has grown to include all kinds of basic protocol components and applications towards the creation of a GNU internet. This is a new major release. Major versions may break protocol compatibility with the 0.26.X versions. Please be aware that Git master is thus henceforth (and has been for a while) INCOMPATIBLE with the 0.26.X GNUnet network, and interactions between old and new peers will result in issues. In terms of usability, users should be aware that there are still a number of known open issues in particular with respect to ease of use, but also some critical privacy issues especially for mobile users. Also, the nascent network is tiny and thus unlikely to provide good anonymity or extensive amounts of interesting information. As a result, the 0.27.0 release is still only suitable for early adopters with some reasonable pain tolerance . Download links [gnunet-0.27.0.tar.gz](http://ftp.gnu.org/gnu/gnunet/gnunet-0.27.0.tar.gz) (signature) [gnunet-fuse-0.27.0.tar.gz](http://ftp.gnu.org/gnu/gnunet/gnunet-fuse-0.27.0.tar.gz) (signature) The GPG key used to sign is: 3D11063C10F98D14BD24D1470B0998EF86F59B6A Note that due to mirror synchronization, not all links might be functional early after the release. For direct access try <http://ftp.gnu.org/gnu/gnunet/> Changes A detailed list of changes can be found in the git log, the NEWS. Known Issues There are known major issues with the TRANSPORT subsystem. There are known moderate implementation limitations in CADET that negatively impact performance. There are known moderate design issues in FS that also impact usability and performance. There are minor implementation limitations in SET that create unnecessary attack surface for availability. The RPS subsystem remains experimental. In addition to

this list, you may also want to consult our bug tracker at bugs.gnunet.org which lists about 190 more specific issues. Thanks This release was the work of many people. The following people contributed code and were thus easily identified: Christian Grothoff, Florian Dold, TheJackiMonster, and Martin Schanzenbach.

- [hello @ Savannah: hello-2.12.3 released \[stable\]](#) (2026/03/18 03:46)

This is to announce hello-2.12.3, a stable release. GNU hello is a demonstration and model of the GNU coding standards for hackers, and a simple example for users. There have been 18 commits by 2 people in the 43 weeks since 2.12.2. See the NEWS below for a brief summary. Thanks to everyone who has contributed! The following people contributed changes to this release: Collin Funk (16) Reuben Thomas (2) Collin [on behalf of the hello maintainers] ===== Here is the GNU hello home page: <https://gnu.org/s/hello/> Here are the compressed sources and a GPG detached signature: <https://ftpmirror.gnu.org/hello/hello-2.12.3.tar.gz> <https://ftpmirror.gnu.org/hello/hello-2.12.3.tar.gz.sig> Use a mirror for higher download bandwidth: <https://www.gnu.org/order/ftp.html> Here are the SHA256 and SHA3-256 checksums: SHA256 (hello-2.12.3.tar.gz) = DV9gFUOC/uELEUocNOeF2LH0kgc64tOm97FHaHs2aqA= SHA3-256 (hello-2.12.3.tar.gz) = VQz4Y71rvDa2iSh59ZUTHiT0wjmFWKo4VcUvpkRi4Ek= Verify the base64 SHA256 checksum with 'cksum -a sha256 --check' from coreutils-9.2 or OpenBSD's cksum since 2007. Verify the base64 SHA3-256 checksum with 'cksum -a sha3 --check' from coreutils-9.8. Use a .sig file to verify that the corresponding file (without the .sig suffix) is intact. First, be sure to download both the .sig file and the corresponding tarball. Then, run a command like this: `gpg --verify hello-2.12.3.tar.gz.sig` The signature should match the fingerprint of the following key: `pub rsa4096/8CE6491AE30D7D75 2024-03-11 [SC] Key fingerprint = 2371 1855 08D1 317B D578 E5CC 8CE6 491A E30D 7D75 uid [ultimate] Collin Funk <collin.funk1@gmail.com>` If that command fails because you don't have the required public key, or that public key has expired, try the following commands to retrieve or refresh it, and then rerun the 'gpg --verify' command. `gpg --locate-external-key collin.funk1@gmail.com gpg --recv-keys 8CE6491AE30D7D75 wget -q -O- 'https://savannah.gnu.org/project/release-gpgkeys.php?group=hello&download=1' | gpg --import -` As a last resort to find the key, you can try the official GNU keyring: `wget -q https://ftp.gnu.org/gnu/gnu-keyring.gpg gpg --keyring gnu-keyring.gpg --verify hello-2.12.3.tar.gz.sig` This release is based on the hello git repository, available as `git clone https://https.git.savannah.gnu.org/git/hello.git` with commit `89fff19b23e35f0e97072507685c92aaae3d04c7` tagged as v2.12.3. For a summary of changes and contributors, see: <https://gitweb.git.savannah.gnu.org/gitweb/?p=hello.git;a=shortlog;h=v2.12.3> or run this command from a git-cloned hello directory: `git shortlog v2.12.2..v2.12.3` This release was bootstrapped with the following tools: Autoconf 2.72 Automake 1.18.1 Gnulib 2026-03-16 4e11e3d07a79a49eaa9b155c43801bbc1e5bd86e NEWS * Noteworthy changes in release 2.12.3 (2026-03-17) [stable] The manual no longer mentions the -h and -v short options which were removed in release 2.11. Update gnulib for compatibility with glibc-2.43. GNU hello no longer fails to build with BSD implementations of the 'make' command. Previously they would be unable to find a target listed as a dependency of the 'hello' program.

- [texmacs @ Savannah: TeXmacs 2.1.5 released](#) (2026/03/17 13:14)

Hello everyone, We are pleased to announce the release of TeXmacs version 2.1.5 This version uses Qt6 by default, supports very high-definition displays, and introduces new ongoing collaborative editing features. On Windows, TeXmacs is now available on the Microsoft Store. On Linux, we have a new Qt6 ApplImage that maximizes compatibility with GNU Linux distributions. On Mac, we have new universal packages. - Download for

Windows: [https://www.texma ... d/windows.en.html](https://www.texma...d/windows.en.html) - Download for macOS: [https://www.texma ... ad/macosex.en.html](https://www.texma...ad/macosex.en.html) - Download for GNU Linux: [https://www.texma ... oad/linux.en.html](https://www.texma...oad/linux.en.html) Happy writing with TeXmacs! The TeXmacs Team

- [health @ Savannah: GNU Health HIS server 5.0.6 patchset bundle released](#) (2026/03/16 20:10)

Dear community We're happy to announce patchset 5.0.6 for the GNU Health Hospital Information System server. The most relevant component of the patchset are automated tests from Tryton (thank you, Cédric!). You can find this and other patches in the Changelog and from the v5.0.6 tag at Codeberg. As usual, the whole source code can be downloaded from GNU.org. Happy hacking!

- [FSF Events: Free Software Directory meeting on IRC: Friday, March 20, starting at 12:00 EDT \(16:00 UTC\)](#) (2026/03/16 11:41)

Join the FSF and friends on Friday, March 20 from 12:00 to 15:00 EDT (16:00 to 19:00 UTC) to help improve the Free Software Directory.

- [FSF Blogs: To tackle plastic waste, tackle DRM \(retracted\)](#) (2026/03/16 11:15)

- [unifont @ Savannah: Unifont 17.0.04 Released](#) (2026/03/13 21:46)

13 March 2026 Unifont 17.0.04 is now available. This is a minor release aligned with Unicode 17.0.0. This release notably includes separate BDF, PCF, and OpenType font files with 28,000+ Unicode T-source Chinese glyphs created by Kusanagi_Sans and Kao Chen-tung (高振東) in font files beginning with "unifont_t". Many other Chinese glyphs have been added. Also, font/Makefile has been reorganized for more efficient font file building. See the ChangeLog file for details. Download this release from GNU server mirrors at: [https://ftpmirror ... /unifont-17.0.04/](https://ftpmirror.../unifont-17.0.04/) or if that fails, [https://ftp.gnu.o ... /unifont-17.0.04/](https://ftp.gnu.o.../unifont-17.0.04/) or, as a last resort, [ftp://ftp.gnu.org ... /unifont-17.0.04/](ftp://ftp.gnu.org.../unifont-17.0.04/) These files are also available on the unifoundry.com website: [https://unifoundr ... /unifont-17.0.04/](https://unifoundr.../unifont-17.0.04/) Font files are in the subdirectory [https://unifoundr ... 0.04/font-builds/](https://unifoundr...0.04/font-builds/) A more detailed description of font changes is available at [https://unifoundr ... nifont/index.html](https://unifoundr...nifont/index.html) and of utility program changes at [https://unifoundr ... nt-utilities.html](https://unifoundr...nt-utilities.html) Information about Hangul modifications is at [https://unifoundr ... hangul/index.html](https://unifoundr...hangul/index.html) and [http://unifoundry ... l-generation.html](http://unifoundry...l-generation.html) Enjoy! Paul Hardy GNU Unifont Maintainer

- [FSF Blogs: The FSF doesn't usually sue for copyright infringement, but when we do, we settle for freedom](#) (2026/03/13 16:20)

- [FSF Blogs: Discord doesn't deserve your unquestioning trust](#) (2026/03/11 19:30)

- [FSF Events: Free Software Directory meeting on IRC: Friday, March 13, starting at 12:00 EDT \(16:00 UTC\)](#) (2026/03/10 14:57)

Join the FSF and friends on Friday, March 13 from 12:00 to 15:00 EDT (16:00 to 19:00 UTC) to help improve the Free Software Directory.

- [FSF News: Job opportunity: Engineering and Certification Manager at the Free Software Foundation](#) (2026/03/10 12:15)

The Free Software Foundation (FSF), a Massachusetts 501(c)(3) charity with a worldwide mission to promote computer user freedom, seeks a motivated and talented individual to be our new Engineering and Certification Manager. This position is ideally full-time and US-based, but exceptions can be made for a qualified candidate.

- [FSF Blogs: March 15 is the deadline to apply for LibreLocal funding](#) (2026/03/09 20:50)

If you want funding for your meetup, apply before it's too late!

- [pspp @ Savannah: PSPP 2.1.1 has been released](#) (2026/03/06 16:48)

I'm very pleased to announce the release of a new version of GNU PSPP. PSPP is a program for statistical analysis of sampled data. It is a free replacement for the proprietary program SPSS. Changes from 2.1.0 to 2.1.1: Translation updates. Bug fixes in build system and tests. No longer mistakenly labeled as a "test release". Please send PSPP bug reports to bug-gnu-pspp@gnu.org.

- [FSF Events: LibreLocal meetup in New Haven, Connecticut, United States](#) (2026/03/04 19:25)

May 29, 2026 at 17:00 EST

- [FSF Events: LibreLocal meetup in Kyiv, Ukraine](#) (2026/03/04 19:15)

May 2026

- [FSF Events: LibreLocal meetup in Shiraz, Iran](#) (2026/03/04 19:10)

May 28, 2026 at 17:00 IRST

- [pspp @ Savannah: PSPP 2.1.0 has been released.](#) (2026/03/04 18:24)

I'm very pleased to announce the release of a new version of GNU PSPP. PSPP is a program for statistical analysis of sampled data. It is a free replacement for the proprietary program SPSS. Changes from 2.0.1 to 2.1.0: Bug fixes. Translation updates. Please send PSPP bug reports to bug-gnu-pspp@gnu.org.

- [texinfo @ Savannah: Texinfo 7.3 released](#) (2026/03/02 18:54)

We have released version 7.3 of Texinfo, the GNU documentation format. It's available via a mirror (xz is much smaller than gz, but gz is available too just in case): [https://ftpmirror ... exinfo-7.3.tar.xz](https://ftpmirror...exinfo-7.3.tar.xz) [https://ftpmirror ... exinfo-7.3.tar.gz](https://ftpmirror...exinfo-7.3.tar.gz) Please send any comments to bug-texinfo@gnu.org. Full announcement: [https://lists.gnu ... -03/msg00007.html](https://lists.gnu...-03/msg00007.html)

- [FSF Blogs: February GNU Spotlight with Amin Bandali featuring nineteen new GNU releases: Nano, Pies, and more!](#) (2026/03/02 16:55)

- [GNU Guix: The 64-bit Hurd is Here!](#) (2026/03/01 10:00)

Fifteen months have passed since our last Guix/Hurd on a Thinkpad X60 post and a lot has happened with respect to the Hurd. And most of you will have guessed, unless you skipped the title of this post, the rumored x86_64 support has landed in Guix! Here is a not-so-short overview of our Hurd work over the past 1.5 years: The build daemon fails when invoking guix authenticate on the Hurd bug was fixed. This was our most pressing problem as it meant that we could not keep our substitutes up to date. It took 15 comments and 13 weeks to get it resolved. Phew! Installer support for (cross)-installing the Hurd. Also adding developer support for running the installer directly from the source tree; Guix 1.5.0 lets you install the Hurd on bare metal. Fix tests in the Shepherd. Update hurd to 0.9.git20250420, gnumach to 1.8+git20250304. Add support for a cross-built gnumach, allowing the removal of an ugly workaround when cross-building for the Hurd. Update rumpkernel to 0-20250111. Support for different childhurd types, a.k.a. 64-bit childhurds in da house. The syslogd used by default is now from the Shepherd streamio, gnumach, and the Shepherd, to make the kernel log work. Update hurd to 0.9.git20251029, gnumach: to 1.8+git20250731. Now that the go-team branch has been merged, gccgo now works (native only). Fix proc server for zombie processes which caused a shepherd test to fail. Fix all the dependencies of the guix package, again: libgit2 tests, dbus, opensp, po4a, Resurrect password hashing. Installer: Fixes for the Hurd. Installer: More clearly mark the Hurd as experimental. Installer: Add Hurd x86_64 as an option. This took 15 comments, uncovering and fixing several bugs. Add support for x86_64-gnu, aka the 64-bit Hurd. The initial patch set consisted of 31 patches. This patch set took four iterations and 208 messages before its final 58 patches were merged to `core-packages-team`. Janneke writes: "Lo and behold, the 64-bit Hurd boots! Again, thanks to the help from the kind folks over at libera #hurd and their excellent work. Do something like: `./pre-inst-env guix system image --image-type=hurd64-qcow2 \ gnu/system/examples/bare-hurd64.templ` Pushed a `core-packages-team` with (this one) GCC 14 commit. Let the fun begin :) We had a lot of fun... Request for merging "core-packages-team" branch: 247 commits, took 114 comments 8 weeks and 24 iterations with 247 commits from 9 people before presenting the initial merge. The actual merge "core-packages-team": 85 more commits to a total of 332, by 17 people and 27 weeks before actual merge. 173 packages with build fixes to relax GCC 14's strictness, 109 package updates to fix build with GCC 14. With this all

in place we can have ci build a 64-bit hurd image, andReport what packages still need to be fixed for that image to build.For convenience we added i586-pc-gnu and x86_64-pc-gnu cross toolchains.Summarizing, building the Guix manifest for the 32-bit Hurd (i586-gnu) should work really well. Sadly, for the 64-bit Hurd (x86_64-gnu) is still a bit problematic as some tests in e.g., openssl, python, cmake, hang. This is still under investigation.What Took You So Long?We're so glad you asked! Usually, adding a new architecture should just take a couple of commits:Add cross-compilation support for the x86_64-pc-gnu target, aka 64-bit Hurd, and thenAdd support for x86_64-gnu, aka the 64-bit Hurd.pretty neat, right? So, what's the story with the 64-bit Hurd? There are two problems: 64-bit Hurd support was added in GCC 14, while Guix was still at GCC 11. This means we "only" had toUpdate the gcc cross compiler to GCC 14 (one, simple commit), andFix all cross builds (initially "just" 23 commits).The second step involves building for all architectures and fixing all breakage. Sometimes, fixing one architecture breaks another.When Guix supported cross-building with GCC 14, and supported the 64-bit Hurd, we could create and boot a 64-bit childhurd. After that, we could start building 64-bit Hurd packages...but only after alsoUse gcc-14, gcc-toolchain-14 on the 64-bit HurdThis, however does not support offloading. For that, we would need to:Update gcc, gcc-toolchain, libgccjit to 14, andMake sure that all packages in commencement.scm successfully build natively on x86_64-hurd, which took only some 35 commits.This can simply be verified by building the hello package:guix build --system=x86_64-gnu helloHowever, GCC 14 is not a regular update: it is waaay more strict with respect to C code compilation. This means that, before actually switching, we had to fix 173 package builds and update another 109 packages to not break all of Guix. This took a total of 17 people and 35 weeks to complete.You can understand that we are excited that the NLnet Foundation has been sponsoring this work!Installing and Using the 64-bit HurdEasiest is to change your 32-bit childhurd definition into 64-bit, by adding(type 'hurd64-qcow2)to your hurd-vm-configuration. And if you don't have a hurd-vm-configuration yet?. Easy, in that case just add(use-service-modules virtualization) [...] (hurd-vm-configuration (type 'hurd64-qcow2))into your your hurd-vm-service-type definition[^0]. And if you don't have a hurd-vm-service-type yet? Easy, in that case just add(use-service-modules virtualization) [...] (service hurd-vm-service-type (hurd-vm-configuration (type 'hurd64-qcow2)))to your operating system definition. Reconfigure your system and you'd be able to:(if you don't have a childhurd definition in your ~/.ssh/config you will have to use something like: ssh -p 10022 root@localhost[^1]).And if you don't have a Guix operating system definition...The 64-bit Hurd is now an option in the installer:and can be installed in a VM. Make sure to use --machine q35 with qemu.To build a disk image for a virtual machine, do:./pre-inst-env guix system image --image-type=hurd64-qcow2 \ gnu/system/examples/bare-hurd64.tmpYou may run it like so:guix shell qemu -- qemu-system-x86_64 -m 2048 -M q35 \ --enable-kvm \ --device e1000,netdev=net0 \ --netdev user,id=net0,hostfwd=tcp:127.0.0.1:10022-:2222 \ --snapshot \ --hda /gnu/store/...-disk-image(note that the 64-bit Hurd does not seem to show a login prompt)and use it like:ssh -p 10022 root@localhost guix build -e '@@ (gnu packages commencement) gnu-make-boot0)'or even, if you build the image with at least --image-size=3G:guix build helloRumpNET SupportUpstream has added support for Intel i8254x Gigabit Ethernet using RumpNET.Damien Zammit wrote:This adds a working rump driver for /dev/wmX cards, which are Intel i8254x Gigabit Ethernet devices. (See man.netbsd.org for "wm(4)") This should be easily extended to support other NICs by contributing some makefile foo to netbsd/rump.Example usage[^2]:settrans -fgap /dev/rumpnet /hurd/rumpnet settrans -fgap /dev/wm0 /hurd/devnode -M /dev/rumpnet wm0 settrans -fgap /servers/socket/2 /hurd/pfinet -i /dev/wm0 ifup /dev/wm0With our updated hurd and rumpkernel packages, this should be available in Guix now too. Please let us know if you got it to work! (If you tried and didn't get it to work, we'd also like to know!)StatusOne of the most frequently asked questions is probably: Does X work on the Hurd yet? The canonical answer to that question is: Please read the GNU/Hurd FAQ.A good summary of the current status was presented by Samuel Thibault in his GNU/Hurd progress at FOSDEM'26, in which he also makes compelling arguments for the Hurd, such as:

Freedom from the system administrator and sharing the GNU heritage and values it's no coincidence that Guix also solves a part of that problem, allowing any user to install packages. Debian GNU/Hurd has been a reality for some years now, reaching 75% of Debian packages being available for the Hurd. As a comparison, in Guix only about 1.7% (32-bit) and 0.9% (64-bit) of packages are available for the Hurd. These percentages fluctuate a bit but continue to grow (both grew with a couple tenth percent point during the preparation of this blog post), and as always, might grow faster with your help. So while Guix GNU/Hurd has an exciting future, please be aware that it lacks many packages and services, including Xorg. If you would simply like to install the Hurd on bare metal running your favorite window manager (e.g.: i3, icewm, etc.) or lightweight desktop environment (Xfce) right now, then installing Debian GNU/Hurd is a good choice. Though we hope to catch up to them soon! Last October, the 64-bit Hurd was reported to run on bare metal. Now that Guix 1.5.0's installer also lets you install the Hurd on bare metal, we'd be thrilled to year from you if you manage to replicate this! What's Next? In an earlier post we tried to answer the question "Why bother with the Hurd anyway?" An obvious question because it is all too easy to get discouraged, to downplay or underestimate the potential social impact of GNU and the Hurd. Echoing Samuel Thibault's talk we would like to add: because it offers a better: Freedom #0: the freedom to run the program as you wish, for any purpose. Freedom from the System Administrator. guix pull is known to work but only by pulling from a local branch doing something like: `mkdir -p src/guix cd src/guix git clone https://git.guix.gnu.org/guix.git master cd master git branch keyring origin/keyring guix pull --url=$HOME/src/guix/master` kinda like we did it in the old days. Other interesting tasks for Guix include: Have guix pull from a non-local URL work on the Hurd, Have guix system reconfigure work on the Hurd, Figure out WiFi support with NetDDE (and add it to installer!), Figure out WiFi support with RumpNET (and add it to installer!), An isolated build environment (or better wait for, err, contribute to the Guile build daemon?), An installer running the Hurd, and, Packages, packages, packages! We tried to make Hurd development as easy and as pleasant as we could. As you have seen, things start to work pretty nicely and there is still plenty of work to do in Guix. In a way this is "merely packaging" the amazing work of others. Some of the real work that needs to be done and which is being discussed and is in progress right now includes: Audio support (this was sponsored by NLnet, thanks!), RumpNET, SMP, Journaling for ext2, AArch64, RISC-V. With the exception maybe of adding RumpNET NICs, these tasks look daunting, and indeed that's a lot of work ahead. But the development environment is certainly an advantage. Take an example: surely anyone who's hacked on device drivers or file systems before would have loved to be able to GDB into the code, restart it, add breakpoints and so on—that's exactly the experience that the Hurd offers. As for Guix, it will make it easy to test changes to the micro-kernel and to the Hurd servers, and that too has the potential to speed up development and make it a very nice experience. SMP support for the 64-bit Hurd During the preparation of this blog post a patch set fixing SMP for the 64-bit Hurd, (well, gnumach actually) was presented by Damien Zammit. So most probably we'll have 64-bit multiprocessing real soon now! It seems however, that we will need new bootstrap binaries for that. Join #guix and #hurd on libera.chat or the mailing lists and get involved! Footnotes[0]: Note: with an up-to-date guix this is no longer necessary! Actually, as the 64-bit Hurd uses rumpdisk exclusively, and gnumach by default uses still its builtin IDE drivers, we also need to tell gnumach about that by adding the (kernel-arguments '("noide")).(use-service-modules virtualization) [...] (hurd-vm-configuration (type 'hurd64-qcow2) (os (operating-system (inherit %hurd-vm-operating-system) (kernel-arguments '("noide"))))) We expect this to be the the default in the future.[1]: You may have to override your childhurd's openssh-service definition, something like (services (modify-services (operating-system-user-services %hurd-vm-operating-system) (openssh-service-type config => (openssh-configuration (inherit config) (authorized-keys `(("root" ,(local-file "/home/janneke/.ssh/janneke.pub")))))))) but you can also take inspiration from the bare-hurd64.tpl template.[2]: Note that while it comes straight from a commit to the Hurd git repository, this is a Debian-specific recipe, Guix does not have it, and per this updated wiki page there's

probably extra networking interface configuration needed too (in Debian you're instructed to -- imperatively -- edit /etc/network/interfaces).

- [GNU MediaGoblin: MediaGoblin 0.15.0](#) (2026/02/26 01:25)

We're pleased to announce the release of GNU MediaGoblin 0.15.0. See the release notes for full details and upgrading instructions. This is a relatively small release to resolve installation issues on Debian Trixie and Bookworm. This version has been tested on Debian Bookworm (12), Debian Trixie (13), Ubuntu 22.04, Ubuntu 24.04 and Fedora 43. This release drops support for Debian Bullseye (11) and Ubuntu 20.04. To join us and help improve MediaGoblin, please visit our getting involved page.

- [FSF News: The FSF announces global call for FSF's LibreLocal 2026 meetups](#) (2026/02/24 22:09)

BOSTON, Massachusetts, USA (Tuesday, February 24, 2026), â€” The Free Software Foundation (FSF) has just launched its global call for LibreLocal 2026.

- [parallel @ Savannah: GNU Parallel 20260222 \('Epstein files'\) released \[stable\]](#) (2026/02/22 22:29)

GNU Parallel 20260222 ('Epstein files') has been released. It is available for download at: `lbry://@GnuParallel:4` Quote of the month: Und die Tage jetzt hab ich GNU parallel für mich entdeckt, auch ne nette Geschichte, gerade wenn's irgendwelche remote APIs sind. -- Vince @dd1des.bsky.social New in this release: No new features. Bug fixes. GNU Parallel - For people who live life in the parallel lane. If you like GNU Parallel record a video testimonial: Say who you are, what you use GNU Parallel for, how it helps you, and what you like most about it. Include a command that uses GNU Parallel if you feel like it. About GNU Parallel GNU Parallel is a shell tool for executing jobs in parallel using one or more computers. A job can be a single command or a small script that has to be run for each of the lines in the input. The typical input is a list of files, a list of hosts, a list of users, a list of URLs, or a list of tables. A job can also be a command that reads from a pipe. GNU Parallel can then split the input and pipe it into commands in parallel. If you use xargs and tee today you will find GNU Parallel very easy to use as GNU Parallel is written to have the same options as xargs. If you write loops in shell, you will find GNU Parallel may be able to replace most of the loops and make them run faster by running several jobs in parallel. GNU Parallel can even replace nested loops. GNU Parallel makes sure output from the commands is the same output as you would get had you run the commands sequentially. This makes it possible to use output from GNU Parallel as input for other programs. For example you can run this to convert all jpeg files into png and gif files and have a progress bar: `parallel --bar convert {1} {1}. {2} ::: *.jpg ::: png gif` Or you can generate big, medium, and small thumbnails of all jpeg files in sub dirs: `find . -name '*.jpg' | parallel convert -geometry {2} {1} {1//}/thumb{2}_{1/} ::: - ::: 50 100 200` You can find more about GNU Parallel at: <http://www.gnu.org/s/parallel/> You can install GNU Parallel in just 10 seconds with: `$ (wget -O - pi.dk/3 || lynx -source pi.dk/3 || curl pi.dk/3/ || \ fetch -o - http://pi.dk/3) > install.sh $ sha1sum install.sh | grep c555f616391c6f7c28bf938044f4ec50 12345678 c555f616 391c6f7c 28bf9380 44f4ec50 $ md5sum install.sh | grep 707275363428aa9e9a136b9a7296dfe4 70727536 3428aa9e 9a136b9a 7296dfe4 $ sha512sum install.sh | grep b24bfe249695e0236f6bc7de85828fe1f08f4259 83320d89 f56698ec 77454856 895edc3e aa16feab 2757966e 5092ef2d 661b8b45 b24bfe24 9695e023 6f6bc7de 85828fe1 f08f4259 6ce5480a 5e1571b2 8b722f21 $ bash install.sh` Watch the intro video on <http://www.youtube.com/watch?v=L284C9FF2488BC6D1> Walk through the tutorial (man parallel_tutorial). Your command line will love you for it. When using programs that use GNU Parallel to process data for publication please cite: O. Tange (2018): GNU Parallel 2018, March 2018, <https://doi.org/10.1146014>. If you like GNU Parallel: Give a demo at your local user group/team/colleagues Post the intro videos on [Reddit/Diaspora*/forums/blogs/Identi.ca/Google+/Twitter/Facebook/LinkedIn/](#) mailing lists Get the merchandise <https://gnuparall.com/igns/gnu-parallel> Request or write a review for your favourite blog or magazine Request or build a package for your favourite distribution (if it is not already there) Invite me for your next

conference If you use programs that use GNU Parallel for research: Please cite GNU Parallel in you publications (use --citation) If GNU Parallel saves you money: (Have your company) donate to FSF [https://my.forg/donate/](https://my.f... .org/donate/) About GNU SQL GNU sql aims to give a simple, unified interface for accessing databases through all the different databases' command line clients. So far the focus has been on giving a common way to specify login information (protocol, username, password, hostname, and port number), size (database and table size), and running queries. The database is addressed using a DBURL. If commands are left out you will get that database's interactive shell. When using GNU SQL for a publication please cite: O. Tange (2011): GNU SQL - A Command Line Tool for Accessing Different Databases Using DBURLs, ;login: The USENIX Magazine, April 2011:29-32. About GNU Niceload GNU niceload slows down a program when the computer load average (or other system activity) is above a certain limit. When the limit is reached the program will be suspended for some time. If the limit is a soft limit the program will be allowed to run for short amounts of time before being suspended again. If the limit is a hard limit the program will only be allowed to run when the system is below the limit.

- [GNU Guix: Result of Sustain and Strengthen Fundraising](#) (2026/02/17 11:00)

Results from Guix Fundraising We're on course to beat our fundraising target to sustain and strength Guix. We're bringing the fundraising campaign to an end, so let's cover how much we've raised and what it means for GNU Guix. After four months of fundraising we've raised €11,378 for the GNU Guix project. This means we've received money for 75% of our €15,000 annual goal. We also pre-registered tickets for Guix Days this year. Pjotr Prins and Manolis Ragkousis have done a stellar job organising it for many years, along with the Declarative and Minimalistic Computing devroom at FOSDEM (videos are up!). Guix Foundation financially supports it as it's a great opportunity for people to spend time together working on improving Guix. Operating a registration system was very successful, raising €3,830 which really contributed to covering the event's costs. Thank you everyone who took part! Recurring donations are critical for the Guix project to be sustainable. If we're certain that there's a regular stream of donations then we can match it with the recurring costs the project incurs (e.g our build farm). This means there's a lot less risk that we'll suddenly have to reduce the shared resources the project depends on: this is where we were last year when we were weeks away from needing to reduce the hosting. Between Stripe and Open Collective 136 people have stepped forward to support the project with recurring donations. During December and January, 17 new people started regular donations. As we'd expect some people stop donating after a while, over that same period we lost 8 recurring donors. The total recurring monthly donations are €1,650. If we annualise those figures then we could raise about €19,800 for the Guix project this year. This doesn't account for any churn, but nonetheless that's fantastic! The impact of recurring donations is considerable as it means a small amount per month really adds up over time. The maths is simple, but don't underestimate how much it helps! The more donations we gather, the more we can do to support Guix. If you'd like to help out the project whether with a single donation or a recurring donation you can: [DONATE NOW](#) SUSE Cares Donation In December SUSE contacted us to tell us that they'd like to donate \$500 to Guix Foundation on behalf of SUSE Cares their philanthropic giving programme. This is an employee programme that enables SUSE employees to support charities of their choice. Tanguy and I have completed the registration documents and we expect to receive the donation shortly. This is fantastic, Thank you SUSE team! Having some support from organisations that use Guix or are aligned with our mission would be great. If you know of an organisation, company or non-profit that might be able to support Guix please get in contact with me. What we've learnt If we take the donations we've received so far, add the registrations from Guix Days and we make a conservative forecast on how recurring donations will come through then we will raise €33,900 for Guix over the year. That's over twice the target we set! That's great and thanks to everyone who's helped Guix. It's been fantastic seeing so many people answer the call to take action and help the project. Guix Foundation has

grown with nearly 100 people joining. This gives us a healthy, user-supported non-profit around Guix. How we're using the money The first priority for using the money we've raised is to support and improve the key infrastructure that the project relies on. One way we'll be doing that is by Guix Foundation joining Codeberg e.V. and financially supporting their efforts. This is important for Guix both because their mission of creating a Free Software platform for collaboration aligns with our goals, but also because we directly rely on Codeberg being able to run a reliable development service. As we know running infrastructure is complex and expensive. Guix Foundation also aims to support the development of Guix, and the community around it. That could mean sponsoring development, running events and adding community services. For Guix Days I put together a talk about the fundraising and our future plans. The talk's available as a PDF, or there's a video on YouTube(1440p) and TILvids Peertube (1080p).

- [Jose E. Marchesi: First package written in Algol 68 lands in Gentoo](#) (2026/02/05 10:00)

To my knowledge Gentoo just became the first GNU/Linux distro ever packaging and distributing a program that happens to be written in Algol 68... have no doubt, others will follow shortly ;) <https://packages.gentoo.org/packages/dev-util/godcc>

- [Jose E. Marchesi: godcc 1.0 released](#) (2026/02/04 21:00)

I am happy to announce the first release of godcc, version 1.0. The tarball [godcc-1.0.tar.gz](https://jemarch.net/godcc-1.0.tar.gz) is now available at <https://jemarch.net/godcc-1.0.tar.gz>. godcc (<https://jemarch.net>) is a full-fledged command-line interface to Compiler Explorer instances such as <https://godbolt.org>. It currently supports getting listings, compiling source files and formatting sources. Happy godccing!

- [coreutils @ Savannah: coreutils-9.10 released \[stable\]](#) (2026/02/04 12:58)

This is to announce coreutils-9.10, a stable release. Notable changes include: - Options in man pages link directly into the full web docs - timeout(1) now kills the command for all terminating signals - paste(1) is now multi-byte character aware - cp(1) fixes an unlikely infinite loop introduced in v9.9 - The multi-call binary is 3.2% smaller There have also been many bug fixes and other changes as summarized in the NEWS below. There have been 288 commits by 10 people in the 12 weeks since 9.9. Thanks to everyone who has contributed! The following people contributed changes to this release: Bernhard Voelker (1) Bruno Haible (1) Christopher Illarionova (2) Collin Funk (92) Dmitry V. Levin (1) Egmont Koblinger (3) Paul Eggert (14) Pádraig Brady (159) Sylvestre Ledru (5) oech3 (10) Pádraig [on behalf of the coreutils maintainers] ===== Here is the GNU coreutils home page: <https://gnu.org/s/coreutils/> Here are the compressed sources: <https://ftp.gnu.org/gnu/coreutils/coreutils-9.10.tar.gz> (15MB) <https://ftp.gnu.org/gnu/coreutils/coreutils-9.10.tar.xz> (6.3MB) Here are the GPG detached signatures: <https://ftp.gnu.org/gnu/coreutils/coreutils-9.10.tar.gz.sig> <https://ftp.gnu.org/gnu/coreutils/coreutils-9.10.tar.xz.sig> Use a mirror for higher download bandwidth: <https://www.gnu.org/order/ftp.html> Here are the SHA256 and SHA3-256 checksums: SHA256 (coreutils-9.10.tar.gz) = 4L3h+2hQIEf8cjyUX6KjH+kZ2mRm7dJDtNqoukjhWI= SHA3-256 (coreutils-9.10.tar.gz) = ajdC0yoxKq5sDXyeL9nMXNSZ26du/3QtZCEo4PNZZkA= SHA256 (coreutils-9.10.tar.xz) = FINamt8LEANzZOLWEqrT2fTso6NEIjztdNEvr0vVHSU= SHA3-256 (coreutils-9.10.tar.xz) = jUv9Ki9gdL5VuXEhDhGyuR+Md4r2PankJ9JCw1xdoWY= Verify the base64 SHA256 checksum with 'cksum -a sha256 --check' from coreutils-9.2 or OpenBSD's cksum since 2007. Verify the base64 SHA3-256 checksum with 'cksum -a sha3 --check' from coreutils-9.8. Use a .sig file to verify that the corresponding file (without the .sig suffix) is intact. First, be sure to download both the .sig file and the corresponding tarball. Then, run a command like this: `gpg --verify coreutils-9.10.tar.xz.sig` The signature should match the fingerprint of the following key: pub rsa4096/0xDF6FD971306037D9 2011-09-23 [SC] Key fingerprint = 6C37 DC12 121A 5006 BC1D B804 DF6F D971 3060 37D9

uid [ultimate] Pádraig Brady <P@draigBrady.com> uid [ultimate] Pádraig Brady <pixelbeat@gnu.org> If that command fails because you don't have the required public key, or that public key has expired, try the following commands to retrieve or refresh it, and then rerun the 'gpg --verify' command. `gpg --locate-external-key P@draigBrady.com gpg --recv-keys DF6FD971306037D9 wget -q -O- 'https://savannah.gnu.org/project/release-gpgkeys.php?group=coreutils&download=1' | gpg --import -` As a last resort to find the key, you can try the official GNU keyring: `wget -q https://ftp.gnu.org/gnu/gnu-keyring.gpg gpg --keyring gnu-keyring.gpg --verify coreutils-9.10.tar.xz.sig` This release is based on the coreutils git repository, available as `git clone https://https.git.savannah.gnu.org/git/coreutils.git` with commit `89b2cd58ac895e3fc0d24d8f10e7e4ba132e7fb6` tagged as v9.10. For a summary of changes and contributors, see: <https://gitweb.git.savannah.gnu.org/gitweb/?p=coreutils.git;a=shortlog;h=v9.10> or run this command from a git-cloned coreutils directory: `git shortlog v9.9..v9.10` This release was bootstrapped with the following tools: `Autoconf 2.72.101-9513b Automake 1.18.1 Gnulib 2026-01-24 1c5e0277c2143dd570d8c88f8923eed2afd8e13b Bison 3.8.2 NEWS * Noteworthy changes in release 9.10 (2026-02-04) [stable] ** Bug fixes` `cp`, `install`, and `mv` no longer enter an infinite loop copying sparse files with `SEEK_HOLE`. E.g., this was seen on ext4 when copying sparse files with extents that are being actively updated, and copy offload is not being used. [bug introduced in coreutils-9.9] `'date'` no longer fails with format directives that return an empty string. [bug introduced in coreutils-9.9] `'dd seek=N of=FILE'` no longer continues copying, overwriting FILE if it exists, if `ftruncate` fails. [bug introduced in coreutils-9.1] `du` and `ls` no longer modify strings returned by `getenv`. POSIX says this is not portable. [bug introduced in fileutils-4.1.6] `'fmt'` now correctly diagnoses read errors. Previously `fmt` generated a generic error for any read error. [bug introduced in coreutils-9.0] `md5sum --text` correctly translates CRLF line endings with the MSYS2 runtime. This also applies to the `sha*sum` and `b2sum` utilities. [This bug was present in "the beginning".] `'numfmt'` no longer drops custom suffixes from numbers it cannot fully parse. [bug introduced with numfmt in coreutils-8.21] `'tail -f --pid'` can no longer exit upon receiving a non terminating signal. On older Linux systems it may have failed with "Interrupted system call". [bug introduced in coreutils-7.5] `'timeout'` will now propagate all terminating signals to the monitored command. Previously `'timeout'` could have exited and left the monitored command running. [bug introduced with timeout in coreutils-7.0] `wc` now documents its `--debug` option, currently used to indicate the line count acceleration being used. [bug introduced in coreutils-9.0] When built with ``clang -fno-inline``, memory allocation issues are again handled in a defined manner. Previously programs may have crashed etc. after a failure to allocate memory. [bug introduced in coreutils-9.0] ** New Features `configure` accepts a new `--enable-single-binary=hardlinks` mode to build the selected programs as hard links to a multi-call binary called "coreutils". This augments the existing "symlinks" and "shebangs" modes already supported by the `--enable-single-binary` option. `'stat'` and `'tail'` now know about the "guest-memfd" file system type. `stat -f -c%T` now reports the file system type, and `tail -f` uses polling for this file system. `'tail'` now accepts the `--debug` option, which is currently used to detail the `--follow` implementation being used. `'du'` now supports the short option `-A` corresponding to the existing long option `--apparent-size`, for compatibility with FreeBSD. ** Changes in behavior All commands now markup option names in `--help` and man pages, with bold attributes, and hyperlinks into the online manual on gnu.org. The links can be configured with the `--enable-manual-url` configure option, and the bold highlighting with `--disable-bold-man-page-references`. At runtime all markup can be disabled with the `TERM=dumb` env var value. `'fmt' -w,--width` no longer includes `'\n'` in the width of a line. I.e., the specified width is interpreted to be an `_inclusive_` maximum. `'ls --hyperlink'` now uses more standard format hyperlinks. `'ESC'` (ST) is now used as a delimiter, instead of `'\a'` (BEL). `'ptx' -t` is no longer a no-op, and now sets the default width to 100 columns. `'timeout'` now honors ignored signals and will not propagate them. E.g., `timeout(1)` in a shell backgrounded job, will not terminate upon receiving `SIGINT` or `SIGQUIT`, as these are ignored by default in shell

background jobs. 'timeout -v -s 0' now prints the signal number 0 instead of EXIT. The multi-call binary now only processes --help or --version options if it is installed with a name ending with "coreutils". This allows for more consistent handling of these options with unsupported commands. ** Improvements The multi-call binary built with configure --enable-single-binary is reduced in size by 3.2% through the more efficient reuse of the cksum utility by the md5sum and sha*sum utilities. 'cksum' now validates its options more consistently. E.g., `cksum --text --tag` now fails like `cksum --tag --text` already did. 'cksum', 'du', and 'wc' now exit promptly upon receiving a write error, which is significant when processing many input files. csplit, ls, and sort, now handle a more complete set of terminating signals. 'du' now processes directories with 10,000 or more entries up to 9 times faster on the Lustre file system. 'paste' now supports multi-byte --delimiters characters. 'pinky' will now exit immediately upon receiving a write error, which is significant when reading large plan or project files. 'readlink' and 'realpath' will now exit promptly upon receiving a write error, which is significant when canonicalizing multiple file names longer than PATH_MAX. 'timeout' on Linux will always terminate the child in the case where the timeout process itself dies, like when it receives a KILL signal for example. ** Build-related Programs now port to C23 platforms that strictly check types when qualifier-generic functions like strchr are used. 'chcon' and 'runcon' stub binaries will be built on systems without libselinux, when configured using --with-selinux. 'kill' and 'uptime' are no longer built by default. These programs can be built with the --enable-install-program=kill,uptime configure option.

- [gettext @ Savannah: GNU gettext 1.0 released](#) (2026/01/29 17:30)

Download from <https://ftp.gnu.org/ftp.gnu.org/pub/gnu/text/ttext-0.26.tar.gz> New in this release: Improvements for maintainers and distributors: In a po/ directory, the PO files are now exactly those that the translators submitted or committed in version control, or a translation project's daemon committed on behalf of the translators. They are no longer regularly updated with respect to the POT file in the same directory. The advantage for maintainers is that the maintainer may commit the PO files in version control, without getting lots of modified files shown by "git status", frequent merge conflicts when merging between branches, a voluminous version control history. The advantage for distributors is that the role of files in a release tarball are clearer: The PO files are source code, whereas the POT file and the *.gmo files are generated files. ATTENTION translators! Translators who work directly on a package's source code (without going through a translation project) now need to run "msginit" before starting work on a PO file. A new program 'po-fetch' is provided, that fetches the translated PO files from a translation project's site on the internet, and updates the LINGUAS file accordingly. In a po/ directory, a new script 'fetch-po' is now added by 'gettextize'. It provides the standard interface for fetching the translated PO files. It typically either invokes the 'po-fetch' program or does nothing. Improvements for translators: msginit: When the PO file already exists, 'msginit' now updates it w.r.t. the POT file, like 'msgmerge' would do. Previously, 'msginit' failed with an error message in this situation. Pretranslation: Two new programs, 'msgpre' and 'spit', are provided, that implement machine translation through a locally installed Large Language Model (LLM). 'msgpre' applies to an entire PO file, 'spit' to a single message. The documentation has a new chapter "Pretranslation". Improvements for maintainers: xgettext: The refactoring suggestion when a translatable string contains an URL or email address can now be inhibited through a command-line option '--no-check=url' or '--no-check=email', or through a comment in the source code of the form `/* xgettext: no-url-check */` or `/* xgettext: no-email-check */` Programming languages support: OCaml: xgettext now supports OCaml. 'msgfmt -c' now verifies the syntax of translations of OCaml format strings. A new example 'hello-ocaml' has been added. Rust: xgettext now recognizes 'gettextrs::gettext' invocations, like 'gettext' invocations. libgettextpo library: The function 'po_message_get_format' now supports distinguishing whether a negative format string mark, such as 'no-c-format', is set or not. The new functions po_message_has_workflow_flag, po_message_set_workflow_flag, po_message_workflow_flags_iterator, po_flag_next, po_flag_iterator_free can be used to manipulate or inspect

the workflow flags of a message. The new functions `po_message_has_sticky_flag`, `po_message_set_sticky_flag`, `po_message_sticky_flags_iterator`, `po_flag_next`, `po_flag_iterator_free` can be used to manipulate or inspect the sticky flags of a message. Emacs PO mode: Restore syntax highlighting in Emacs version 30 or newer.

- [GNU Artanis: Techical report 2026-Jan-28](#) (2026/01/28 08:26)

- [parallel @ Savannah: GNU Parallel 20260122 \('Maduro'\) released \[stable\]](#) (2026/01/27 23:44)

GNU Parallel 20260122 ('Maduro') has been released. It is available for download at: `lbry://@GnuParallel:4` Quote of the month: 64コアで、64並列でsimulationを回してtopコマンドで状況を見るのは心地よい。簡単に並列処理を実現できるGNU parallelコマンドは素晴らしい☺ -- Daisuke lizuka @diizuka@twitter New in this release: No new features. Bug fixes. GNU Parallel - For people who live life in the parallel lane. If you like GNU Parallel record a video testimonial: Say who you are, what you use GNU Parallel for, how it helps you, and what you like most about it. Include a command that uses GNU Parallel if you feel like it. About GNU Parallel GNU Parallel is a shell tool for executing jobs in parallel using one or more computers. A job can be a single command or a small script that has to be run for each of the lines in the input. The typical input is a list of files, a list of hosts, a list of users, a list of URLs, or a list of tables. A job can also be a command that reads from a pipe. GNU Parallel can then split the input and pipe it into commands in parallel. If you use `xargs` and `tee` today you will find GNU Parallel very easy to use as GNU Parallel is written to have the same options as `xargs`. If you write loops in shell, you will find GNU Parallel may be able to replace most of the loops and make them run faster by running several jobs in parallel. GNU Parallel can even replace nested loops. GNU Parallel makes sure output from the commands is the same output as you would get had you run the commands sequentially. This makes it possible to use output from GNU Parallel as input for other programs. For example you can run this to convert all jpeg files into png and gif files and have a progress bar: `parallel --bar convert {1} {1.}.{2} ::: *.jpg ::: png gif` Or you can generate big, medium, and small thumbnails of all jpeg files in sub dirs: `find . -name '*.jpg' | parallel convert -geometry {2} {1} {1//}/thumb{2}_{1/} ::: - ::: 50 100 200` You can find more about GNU Parallel at: <http://www.gnu.org/s/parallel/> You can install GNU Parallel in just 10 seconds with: `$ (wget -O - pi.dk/3 || lynx -source pi.dk/3 || curl pi.dk/3/ || \ fetch -o - http://pi.dk/3) > install.sh $ sha1sum install.sh | grep c555f616391c6f7c28bf938044f4ec50 12345678 c555f616 391c6f7c 28bf9380 44f4ec50 $ md5sum install.sh | grep 707275363428aa9e9a136b9a7296dfe4 70727536 3428aa9e 9a136b9a 7296dfe4 $ sha512sum install.sh | grep b24bfe249695e0236f6bc7de85828fe1f08f4259 83320d89 f56698ec 77454856 895edc3e aa16feab 2757966e 5092ef2d 661b8b45 b24bfe24 9695e023 6f6bc7de 85828fe1 f08f4259 6ce5480a 5e1571b2 8b722f21 $ bash install.sh` Watch the intro video on <http://www.youtube.com/watch?v=L284C9FF2488BC6D1> Walk through the tutorial (`man parallel_tutorial`). Your command line will love you for it. When using programs that use GNU Parallel to process data for publication please cite: O. Tange (2018): GNU Parallel 2018, March 2018, <https://doi.org/10.1101/281146>. If you like GNU Parallel: Give a demo at your local user group/team/colleagues Post the intro videos on [Reddit/Diaspora*/forums/blogs/Identi.ca/Google+/Twitter/Facebook/LinkedIn/](#) mailing lists Get the merchandise <https://gnuparall.com/igns/gnu-parallel> Request or write a review for your favourite blog or magazine Request or build a package for your favourite distribution (if it is not already there) Invite me for your next conference If you use programs that use GNU Parallel for research: Please cite GNU Parallel in you publications (use `--citation`) If GNU Parallel saves you money: (Have your company) donate to FSF <https://my.fsf.org/donate/> About GNU SQL GNU sql aims to give a simple, unified interface for accessing databases through all the different databases' command line clients. So far the focus has been on giving a common way to specify login information (protocol, username, password, hostname, and port number), size (database and table size), and running queries. The database is addressed using a DBURL. If commands are left out you will get that database's interactive shell. When using GNU SQL for a

publication please cite: O. Tange (2011): GNU SQL - A Command Line Tool for Accessing Different Databases Using DBURLs, ;login: The USENIX Magazine, April 2011:29-32. About GNU Niceload GNU niceload slows down a program when the computer load average (or other system activity) is above a certain limit. When the limit is reached the program will be suspended for some time. If the limit is a soft limit the program will be allowed to run for short amounts of time before being suspended again. If the limit is a hard limit the program will only be allowed to run when the system is below the limit.

- [GNU Guix: GNU Guix 1.5.0 released](#) (2026/01/23 14:00)

We are pleased to announce the release of GNU Guix version 1.5.0! The release comes with ISO-9660 installation images, virtual machine images, and with tarballs to install the package manager on top of your GNU/Linux distro, either from source or from binaries—check out the download page. Guix users can update by running `guix pull`. It's been 3 years since the previous release. That's a lot of time, reflecting both the fact that, as a rolling release, users continuously get new features and update by running `guix pull`; but it also shows a lack of processes, something that we had to address before another release could be made. During that time, Guix received about 71,338 commits by 744 people, which include many new features; the project also got a new decision-making process, migrated to Codeberg and started a fundraising campaign. That's just the surface among so many great changes, so keep reading! Illustration by Luis Felipe, published under CC-BY-SA 4.0. This post provides highlights for all the hard work that went into this release. There's a lot to talk about so make yourself comfortable, relax, and enjoy. Guix ecosystem To start with, the Guix ecosystem has seen many exciting developments to the way we collaborate and make decisions! Firstly, the project adopted with unanimity a new consensus-based decision making process. This process fills a need to be able to gather consensus on significant changes to the project, something that was getting very complicated with the growing number of contributors to the project. Now, the process provides a clear framework for any contributor to propose and implement important changes. These can be submitted as Guix Consensus Documents (GCDs), each GCD goes through the multiple steps of consensus decision making before being accepted or withdrawn. Secondly, using this process, the project was able to collectively migrate to Codeberg. This means that all repositories, and bug trackers are now at the same place on Codeberg and that contributions are now made with pull requests instead of patch series. Thirdly, a new release process was adopted to bring an annual release cycle to Guix. This release is the first to follow this process, with hopefully many others to come! Lastly, a "Planet" website for Guix is now available at <https://planet.guix.gnu.org>. It aggregates blogs from various Guix hackers and contributors to bring you the latest and greatest in Guix news. Stronger distribution Three years is a long time for free and open source software! Enough time for 12,525 new packages and 29,932 package updates to the Guix repository. Here are the best highlights: To start, KDE Plasma 6.5 is now available with the new `plasma-desktop-service-type`! Continuing on desktops; GNOME has been updated from version 42 to 46 and now uses Wayland by default. The `gnome-desktop-service-type` was made more modular to better customize the default set of GNOME applications. Guix System is now using version 1.0 of the GNU Shepherd, which now supports timed services, `kexec` reboot and has new services for system logs and log rotation which are now used by Guix System instead of `Rottlog` and `syslogd`. There are around 40 new system services to choose from, including `Forgejo Runner`, `RabbitMQ`, `iwd`, and `dhcpcd` to name a few. `setuid`-programs has been replaced with `privileged`-programs in operating-system definitions to support giving specific Linux capabilities. Additionally, the `nss-certs` package is now included in `%base`-packages. More than 12,500 packages were added, keeping Guix in the top-ten biggest distributions according to Repology! Among the many noteworthy updates, we now have GCC 15.2.0, Emacs 30.2, Icecat and Librewolf 140, LLVM 21.1.8 and Linux-libre 6.17.12. Team activity In the last release, we introduced structured cooperation using teams. There are now 50 teams distributing the many aspects of the distribution.

We have per-language teams like python, rust and zig ensuring updates for packages and build systems as well as thematic teams like electronics, hpc and bioinformatics working on specific application domains. Here are what some of these teams have been up to: The HPC team published their annual activity report 2024, showing the exciting developments of Guix in High-Performance Computing. The electronics team is maintaining free software based Electronic Design Automation (EDA) packages to cover the needs of professionals and hobbyists in the domain with tools such as KiCad, LibrePCB, Xschem, Qucs-S and Ringdove EDA, as well as Verilog, SystemVerilog and VHDL compilers and a toolchain for programmable designs on GateMate FPGAs. They are also collaborating with the Free Silicon Foundation (F-Si) to push free software in the EDA space! The science team has been able to add a myriad of Astronomy related packages, accompanied by the Python team bringing the move to the new pyproject.toml-based build system as well as the NumPy 2 update. Finally, the rust team created a new packaging model to efficiently package rust crates, and was able to migrate the Rust collection, 150+ packages with 3,600+ libraries, in just under two weeks; making the Rust packaging process much easier for everyone. Full source bootstraps Full-source bootstraps of the Zig and Mono compilers are now available, and the existing bootstrap of Guix has been reduced once again! Full-source bootstraps are Guix's solution to the trusting trust problem: compilers are usually compiled by themselves, so how can you build a compiler without trusting an existing binary? Read these posts to learn more about this fascinating problem: The Full-Source Bootstrap: Building from source all the way down Zig reproduced without binaries Restoring Zig bootstrap chain in Guix (in Traditional Chinese) Adding a fully bootstrapped Mono Improved CLIThe guix graph command has new backends for GraphML and CycloneDX JSON, meaning Guix can now be used to generate complete Software Bill of Material (SBOM) down to the first bootstrap binary! guix shell containers have been improved with a --nesting option to use Guix within the container and a --emulate-fhs option that can be used to run software expecting a Filesystem Hierarchy Standard (FHS) compliant filesystem. The guix pack command also received new backends to create RPM packages and AppImages that can be used to publish your Guix packages to non-Guix users. Lastly, a new guix locate command is now available to find which packages provide a given file. Security improvements It is now possible to run the Guix daemon without root privileges, reducing the impact of privilege escalation vulnerabilities. This "rootless" mode is now the default when installing Guix 1.5.0 on distros other than Guix System; on Guix System, it currently has to be explicitly enabled by setting (privileged? #f) in guix-configuration. Existing installation on distros other than Guix System can also be migrated to "rootless". This is possible thanks to the user namespaces. It might be possible that on your system, the user namespaces are not allowed for guix due to the lack of an AppArmor profile. Because of that, we've also included AppArmor profiles that are installed by default on foreign systems. Finally, the Guix daemon received security fixes for CVE-2024-27297, CVE-2024-52867, CVE-2025-46415, CVE-2025-46416 and CVE-2025-59378. Widened architecture support Release tarballs are now available for the RISC-V 64-bit architecture (riscv64-linux). The x86_64 architecture saw some development as well, with the experimental support of the GNU Hurd kernel (x86_64-gnu), aiming to be another significant step in the adoption and development of the Hurd. Overall support for the Hurd was greatly improved, it is now an option in the installer, childhurd can be automatically created with a system service and it can even run on a Thinkpad X60! Fundraising campaign Surprisingly, making a completely free software distribution does not come for free! The Guix project needs your help to pay the infrastructure costs of build farms, web servers and QA tools that are essential to making this release happen. If you appreciate all of the work that is done to bring you this one-of-a-kind distro: please donate to the Guix Foundation! Acknowledgments For the release, thanks to all the release team members: Rutherford, Rodion Goritskov, Efraim Flashner, and NoÃ© Lopez. Thanks as well to the release helpers: Andreas Enge, Mothacehe, Dariqq and Ludovic CourtÃ©s. For creating the release process, thanks to Steve George. For their Guix contributions, thanks to the 744 wonderful people who contributed and

whose names we don't list here (it would be a bit long). They can be listed with `git log --oneline v1.4.0..v1.5.0 --format="%an" | sort -u`. Every commit counts and is always appreciated. About GNU Guix GNU Guix is a transactional package manager and an advanced distribution of the GNU system that respects user freedom. Guix can be used on top of any system running the Hurd or the Linux kernel, or it can be used as a standalone operating system distribution for i686, x86_64, ARMv7, AArch64, RISC-V and POWER9 machines. In addition to standard package management features, Guix supports transactional upgrades and roll-backs, unprivileged package management, per-user profiles, and garbage collection. When used as a standalone GNU/Linux distribution, Guix offers a declarative, stateless approach to operating system configuration management. Guix is highly customizable and hackable through Guile programming interfaces and extensions to the Scheme language.

- [GNU Guix: Meet Guix at FOSDEM](#) (2026/01/22 13:00)

It's that time of the year again: next week is FOSDEM time! As in previous years, many Guix people will be in Brussels. Right after FOSDEM, about sixty of us will gather on February 2–3 for the Guix Days! First things first: Guix presence at FOSDEM. On Saturday, January 31st: In Name resolution in package management systems — A reproducibility perspective, Gábor Boskovits will look at how several package managers refer to packages and how this affects reproducibility. Simon Tournier will give a lightning talk in the Bioinformatics track, Guixifying workflow management system: past, present, maybe future?, discussing the Guix Workflow Language (GWL), the ccwl, and ravanan. On Sunday, February 1st, the Declarative & Minimalistic Computing track will once again be a Guile & Guix lair. The whole track is amazing, with top-notch talks and speakers; particularly relevant to Guix and Guile hackers are the following: Sergio Pastor Pérez will give a talk entitled BLUE — A generic build system crafted entirely in Guile. There are clear connections with Guix but BLUE could well become the new standard build system for Guile developers! In Modern Development Tools and Practices for GNU Guile Andrew Tropin will talk about live programming at the REPL, in particular with the Ares/Arei interactive development environment. On the same theme, Jessica Talon of Spritely will talk about Guile development outside of Emacs—addressing a real need of Guile and Guix outreach efforts. In Lisp is clay: the power of composable DSLs, the inimitable Christine Lemmer-Webber will talk about this Lisp foundation that makes Spritely and Guix so powerful. David Thompson (also of Spritely!) will talk about Functional reactive programming with propagators, which sounds like an exciting topic for any functional programming person and any programmer who's worked on user interfaces and other kinds of “reactive” programs. Simon Josefsson will talk about Guix Container Images — and what you can do with them, showing how to add Guix container images to registries and how to use them in continuous integration and continuous delivery (CI/CD) pipelines. You can have more Guix bliss on Sunday afternoon: In a talk entitled Package management in the hands of users: dream and reality, I (Ludovic Courtès) will reflect on successes and failures bringing package management to HPC supercomputer users. Samuel Thibault will share Updates on GNU/Hurd progress, which includes Guix goodness and good news for a practical empowering operating system. Guix Days will take place on Monday and Tuesday right after FOSDEM, at our usual venue. Sixty people already registered, which is our maximum capacity—don't just show up and hope for the best. As always, this will be unconference style: we'll make the program as we go, discussing hot topics such as the crowdfunding campaign, an update on Guix Foundation, processes and governance, as well as the more technical topics we're fond of. This year marks the tenth anniversary of the Guile/Declarative & Minimalistic Computing track, and the eighth Guix Days. Shout out to our friends Pjotr Prins and Manolis Ragkousis, who have spearheaded the two events during all these years, and to all the volunteers who helped them on the way! This yearly Brussels gathering has been instrumental in building, shaping, and strengthening our community; to those who can be present, it's the energizing and refreshing moment of the year. To Pjotr, to Manolis: thank you! Guix Days graphics are copyright © 2024 Luis Felipe López Acevedo, under CC-BY-SA 4.0, available from Luis' Guix graphics repository. Picture of “Au Bon

Vieux Temps" sign © 2025 Ludovic Courtès, under CC-BY-SA 4.0.

- [GNU Taler news: Critical analysis of digital euro published](#) (2026/01/13 23:00)

The Springer journal "Digital Finance" has recently published "The proposed design of the digital euro: A critical analysis" by Mikolai Gütschow and Bernd Lucke. They describe serious flaws in the digital euro design as proposed by the European Commission and propose GNU Taler as an alternative technology for a potential CBDC with tangible benefits for Europeans.

- [Simon Josefsson: Debian Libre Live 13.3.0 is released!](#) (2026/01/13 13:53)

Following up on my initial announcement about Debian Libre Live I am happy to report on continued progress and the release of Debian Libre Live version 13.3.0. Since both this and the previous 13.2.0 release are based on the stable Debian trixie release, there really isn't a lot of major changes but instead incremental minor progress for the installation process. Repeated installations has a tendency to reveal bugs, and we have resolved the apt sources list confusion for Calamares-based installations and a couple of other nits. This release is more polished and we are not aware of any known remaining issues with them (unlike for earlier versions which were released with known problems), although we conservatively regard the project as still in beta. A Debian Libre Live logo is needed before marking this as stable, any graphically talented takers? (Please base it on the Debian SVG upstream logo image.) We provide GNOME, KDE, and XFCE desktop images, as well as text-only "standard" image, which match the regular Debian Live images with non-free software on them, but also provide a "slim" variant which is merely 750MB compared to the 1.9GB "standard" image. The slim image can still start a debian installer, and can still boot into a minimal live text-based system. The GNOME, KDE and XFCE desktop images feature the Calamares installer, and we have performed testing on a variety of machines. The standard and slim images does not have a installer from the running live system, but all images support a boot menu entry to start the installer. With this release we also extend our arm64 support to two tested platforms. The current list of successfully installed and supported systems now include the following hardware: Desktop ADLINK Ampere Altra Developer Platform arm64 Neoverse N1 Desktop MSI Z790-P WIFI PRO i9-14900K Dasharo Laptop Framework 13 AMD AI 9 HX 370 Laptop Lenovo X201 i7-620M Laptop NovaCustom NV56 Intel Ultra 7 155H i915 Dasharo Server Dell PowerEdge R630 2xE2680v4 Server/Router Protectli VP2440 Server Supermicro MegaDC ARS-110M-NR Ampere Altra Max 128 core 2x25GBe This is a very limited set of machines, but the diversity in CPUs and architecture should hopefully reflect well on a wide variety of commonly available machines. Several of these machines are crippled (usually GPU or WiFi) without adding non-free software, complain at your hardware vendor and adapt your use-cases and future purchases. The images are as follows, with SHA256SUM checksums and GnuPG signature on the 13.3.0 release page. Amd64 GNOME `debian-live-13.3.0-amd64-libre-gnome.iso` Amd64 KDE `debian-live-13.3.0-amd64-libre-kde.iso` Amd64 XFCE `debian-live-13.3.0-amd64-libre-xfce.iso` Amd64 Standard `debian-live-13.3.0-amd64-libre-standard.iso` Amd64 Slim `debian-live-13.3.0-amd64-libre-slim.iso` Arm64 GNOME `debian-live-13.3.0-arm64-libre-gnome.iso` Arm64 KDE `debian-live-13.3.0-arm64-libre-kde.iso` Arm64 XFCE `debian-live-13.3.0-arm64-libre-xfce.iso` Arm64 Standard `debian-live-13.3.0-arm64-libre-standard.iso` Arm64 Slim `debian-live-13.3.0-arm64-libre-slim.iso` Curious how the images were made? Fear not, for the Debian Libre Live project README has documentation, the `run.sh` script is short and the `.gitlab-ci.yml` CI/CD Pipeline definition file brief. Happy Libre OS hacking!

- [www-zh-cn @ Savannah: Summary 2025](#) (2026/01/07 07:46)

Dear GNU CCT: Here is summary of GNU website from GNU: 2025 had a splash of activity; a few teams who were dormant in 2024 made a notable progress, in terms of new translations or updating the existing ones. General Statistics About 2/3 new translations were made by the Chinese (zh-cn) team this year; then the Greek and Albanian teams followed. The Polish and Dutch teams considerably reduced the amount of

their outdated translations. Currently, the total amount of translations is over 3400; the overall percentage of outdated translations was about 5% lower than in 2024. The table below shows the number and size of newly translated articles in important directories and typical number of outdated GNUified translations throughout the year.

team	new	outdated	el	eo *	es	fr	ml	nl	pl	ru	sq	tr	zh-cn	total
	5 (81.9Ki)	19 (41%)	5 (81.9Ki)	1 (33.6Ki)	4 (34.2Ki)	1 (7.9Ki)	1 (11.2Ki)	0 (0.0Ki)	1 (9.9Ki)	2 (26.9Ki)	4 (73.9Ki)	4 (52.5Ki)	39 (797.9Ki)	62 (1130Ki)
			2.0 (0.9%)	1.4 (0.4%)	2.0 (0.9%)	1.4 (0.4%)	29 (88%)	0 (0.0%)	16 (10%)	2.2 (0.7%)	9.0 (10%)	1.9 (1.4%)	1.3 (0.5%)	

The Esperanto translation was installed by GNU Translation Managers without establishing a new team. For the reference: 2 new articles were added, amounting to 27Ki (vs. 4 articles and 44Ki in 2024); the number of commits (about 500 changes in approximately 90 English files) was almost twice as many as in 2024. Orphaned Teams, New and Reformed Teams No teams were orphaned, and no new teams were established. Greek and Dutch teams changed their status to active without a reorganization. A volunteer requested creating the Georgian team, with no further progress. Thank you for your contribution. I wish you all a happy and successful 2026. Happy hacking. wxie

- [GNU Taler news: P15 CoNetWorking Space accepts GNU Taler payments in eCHF](#) (2026/01/05 23:00)

The P15 CoNetWorking Space in Biel/Bienne right next to the train station (and the BFH) is the first shop to accept GNU Taler payments in Swiss francs (eCHF) issued by Taler Operations AG and thus the first merchant accepting Taler payments in fiat currency. P15 is a great space to network, so go check it out!

- [GNU Taler news: GNU Taler presented at 39C3](#) (2026/01/04 23:00)

Mikolai Gütschow and signum gave a talk at the 39th Chaos Communication Congress (39C3) in Hamburg, Germany, where they reported on their good experiences with offering GNU Taler as a local payment system at LugCamp 2024 and Datenspuren 2024 and 2025.

- [Amin Bandali: The People of Emacs](#) (2025/12/31 14:09)

GNU Emacs has been my primary computing environment of choice for over a decade. Emacs has enabled me to perform a wide array of tasks involving human and computer languages, such as reading and writing notes, emails, chats, programs, and more, all in a cohesive and consistent environment that I can tailor exactly to my needs and liking. Coming from a Vim background, I started my Emacs journey trying some configuration frameworks that provided vi-like key bindings, and after a few Emacs bankruptcies, ended up with my current homegrown configuration that I wrote from scratch gradually over the last 7 years, with inspiration from the configurations of some folks who shared theirs publicly. Though my configuration has been mostly stable for a few years now and I consciously keep the number of external packages I use very small, I occasionally add small bits and pieces to my configuration when I'm inspired after learning about a neat feature or package on the blogs aggregated on Planet Emacslife, the messages sent to the Emacs mailing lists, or the videos from the annual EmacsConf conference. I like getting a glimpse of other people's worlds through the lens of their creative works such as writings, be it prose or Emacs Lisp. That's only possible when people share freely, free as in freedom. I'm thankful to Richard Stallman for his foresight to imbue GNU Emacs with that freedom from the very beginning and for his lifelong fight for computer user freedom, and to the many other folks who have joined the free software movement since then and have fought the good fight. I've been inspired and encouraged by many awesome Emacs people through the years. People like Corwin Brust with his joyful creative energy around Emacs and the road to software freedom, Sacha Chua and her philosophy of

leading a life of learning, sharing, and scaling, Gopar and his enthusiasm for Emacs and its intersection with the Python world, folks like Protesilaos Stavrou and Greg Farrow who discovered Emacs initially as non-programmers yet were enamoured by its embodiment of software freedom in practice and went on to integrate it into their everyday lives, and Shoshin of the Cicadas cooperative at the intersection of humanity and technology sharing his passion for the human element and community by developing and contributing input methods for his ancestral language of Lakota to GNU Emacs. I'm deeply inspired by each of these wonderful people, and grateful for having known them and for each of their unique perspectives and life stories with which they have enriched my experience in Emacs and the free software world. As wonderful and impactful as Emacs has been in the lives of the many who have come to know it throughout the decades that it's been around, it would not have become what it has been, what it is today, and what it may become in the future without its community of passionate users and contributors. The People of Emacs are all of us. Here's to many more of us, enjoying many more years of Emacs and software freedom together even if spread far apart. Take care, and so long for now. Inspired by the Emacs Carnival theme for this month, The People of Emacs. Thanks to George Jones for hosting.

- [FSF News: Eko K. A. Owen joins the FSF board as the union staff pick](#) (2025/12/29 22:45)
BOSTON, Massachusetts, USA (December 29, 2025) – The Free Software Foundation (FSF) announced today that Eko K. A. Owen will follow in Ian Kelling's footsteps by becoming the second union staff-elected board member on the organization's board of directors.
- [Jose E. Marchesi: Gemini capsule capsule.jemarch.net](#) (2025/12/29 19:00)
Recently I have been using Gemini, a sort of a modernized Gopher, more and more, and have finally decided to create and maintain my own Gemini capsule, that you can find at <gemini://capsule.jemarch.net>. The plan, moving forward, is to publish basically the same contents in both www and gemini versions of this homepage. Salud!
- [Jose E. Marchesi: Gemini capsule jemarch.srht.site](#) (2025/12/29 19:00)
Recently I have been using Gemini, a sort of a modernized Gopher, more and more, and have finally decided to create and maintain my own Gemini capsule, that you can find at <gemini://jemarch.srht.site>. The plan, moving forward, is to publish basically the same contents in both www and gemini versions of this homepage. Salud!
- [FSF News: Free Software Foundation receives historic private donations](#) (2025/12/24 22:45)
Boston, Massachusetts, USA (Wednesday, December 24, 2025) -- The Free Software Foundation (FSF) today announced it received two major contributions totaling around \$900,000 USD.
- [parallel @ Savannah: GNU Parallel 20251222 \('Bondi'\) released \[stable\]](#) (2025/12/24 01:34)
GNU Parallel 20251222 ('Bondi') has been released. It is available for download at: <lbry://@GnuParallel:4> Quote of the month: Used? gnu parallel is my new favorite toy -- Eytan Adar [@eytan.adar.prof](mailto:eytan.adar.prof) New in this release: No new features. Bug fixes. GNU Parallel - For people who live life in the parallel lane. If you like GNU Parallel record a video testimonial: Say who you are, what you use GNU Parallel for, how it helps you, and what you like most about it. Include a command that uses GNU Parallel if you feel like it. About GNU Parallel GNU Parallel is a shell tool for executing jobs in parallel using one or more computers. A job can be a single command or a small script that has to be run for each of the lines in the input. The typical input is a list of files, a list of hosts, a list of users, a list of URLs, or a list of tables. A job can also be a command that reads from a pipe. GNU Parallel can then split the input and pipe it into commands in parallel. If you use xargs and tee today you will find GNU Parallel very easy to use as GNU Parallel is written to have the same options as xargs. If you write loops in shell, you will find GNU Parallel may be able to

replace most of the loops and make them run faster by running several jobs in parallel. GNU Parallel can even replace nested loops. GNU Parallel makes sure output from the commands is the same output as you would get had you run the commands sequentially. This makes it possible to use output from GNU Parallel as input for other programs. For example you can run this to convert all jpeg files into png and gif files and have a progress bar: `parallel --bar convert {1} {1.}.{2} ::: *.jpg ::: png gif` Or you can generate big, medium, and small thumbnails of all jpeg files in sub dirs: `find . -name '*.jpg' | parallel convert -geometry {2} {1} {1//}/thumb{2}_{1/} :::: - ::: 50 100 200` You can find more about GNU Parallel at: <http://www.gnu.org/s/parallel/> You can install GNU Parallel in just 10 seconds with: `$ (wget -O - pi.dk/3 || lynx -source pi.dk/3 || curl pi.dk/3/ || \ fetch -o - http://pi.dk/3) > install.sh $ sha1sum install.sh | grep c555f616391c6f7c28bf938044f4ec50 12345678 c555f616391c6f7c 28bf9380 44f4ec50 $ md5sum install.sh | grep 707275363428aa9e9a136b9a7296dfe4 70727536 3428aa9e 9a136b9a 7296dfe4 $ sha512sum install.sh | grep b24bfe249695e0236f6bc7de85828fe1f08f4259 83320d89 f56698ec 77454856 895edc3e aa16feab 2757966e5092ef2d 661b8b45 b24bfe24 9695e023 6f6bc7de 85828fe1 f08f4259 6ce5480a 5e1571b2 8b722f21 $ bash install.sh` Watch the intro video on <http://www.youtub...L284C9FF2488BC6D1> Walk through the tutorial (man parallel_tutorial). Your command line will love you for it. When using programs that use GNU Parallel to process data for publication please cite: O. Tange (2018): GNU Parallel 2018, March 2018, <https://doi.org/10.1111/zenodo.1146014>. If you like GNU Parallel: Give a demo at your local user group/team/colleagues Post the intro videos on [Reddit/Diaspora*/forums/blogs/](#) [Identi.ca/Google+/Twitter/Facebook/Linkedin/](#) mailing lists Get the merchandise <https://gnuparall...igns/gnu-parallel> Request or write a review for your favourite blog or magazine Request or build a package for your favourite distribution (if it is not already there) Invite me for your next conference If you use programs that use GNU Parallel for research: Please cite GNU Parallel in you publications (use --citation) If GNU Parallel saves you money: (Have your company) donate to FSF <https://my.f...org/donate/> About GNU SQL GNU sql aims to give a simple, unified interface for accessing databases through all the different databases' command line clients. So far the focus has been on giving a common way to specify login information (protocol, username, password, hostname, and port number), size (database and table size), and running queries. The database is addressed using a DBURL. If commands are left out you will get that database's interactive shell. When using GNU SQL for a publication please cite: O. Tange (2011): GNU SQL - A Command Line Tool for Accessing Different Databases Using DBURLs, ;login: The USENIX Magazine, April 2011:29-32. About GNU Niceload GNU niceload slows down a program when the computer load average (or other system activity) is above a certain limit. When the limit is reached the program will be suspended for some time. If the limit is a soft limit the program will be allowed to run for short amounts of time before being suspended again. If the limit is a hard limit the program will only be allowed to run when the system is below the limit.

- [GNUet News: GNUet 0.26.2](#) (2025/12/22 23:00)

GNUet 0.26.2 This is a bugfix release for gnet 0.26.1. It fixes some regressions and minor bugs. Links Tarball download (tar.gz) (signature) You can find a detailed list of changes in the git log and noteworthy changes in the NEWS file. The GPG key used to sign is: 3D11063C10F98D14BD24D1470B0998EF86F59B6A Note that due to mirror synchronization, not all links may be functional early after the release. For direct access try <https://ftp.gnu.org/gnu/gnet/>

- [GNU Taler news: GNU Taler 1.3 released](#) (2025/12/22 23:00)

We are happy to announce the release of GNU Taler v1.3.

- [health @ Savannah: GNU Health HIS 5.0.5 - Ready for Python 3.14](#) (2025/12/16 12:46)

Dear community I am happy to announce that since patchset 5.0.5, GNU Health Information system is ready for Python 3.14. All GNU Health HIS

packages have been updated so they allow Python 3.14. The GNU Health GTK client and GNU Health control installation and instance manager have also been upgraded. Operating systems like Void Linux already upgraded to Python 3.14, and now you can enjoy this Pi.thon release in GNUHealth. As usual, backup your database, local modules/packages and GNU Health filesystem before upgrading, and report any issue you may find. For detailed information and changes, please visit the GNU Health HIS repositories at Codeberg (<https://codeberg. ... org/gnuhealth/his>) You can download the source code directly from GNU.org and the packages from PyPI. Happy hacking ♥

- [health @ Savannah: GNU Health HIS server 5.0.4 patchset bundle released](#) (2025/12/13 20:53)

Dear community The patchset 5.0.4 of GNU Health Information System is out! This is a small patch related to the calendar package. If you use the vanilla / standard installation, you can update the server and the dependencies from the gnuhealth control center (<https://docs.gnuh ... ontrolcenter.html>) Backup As usual, before you upgrade your instance, make sure you have made a backup of your DB instance and "attach" resource ! Happy hacking ♥ Changelog for 5.0.4 5bbd80c38: health_calendar: Fix issue #164 - AttributeError when creating work schedule Tue Dec 2 11:13:26 2025 +0000 Luis Falcon For more information, please check our Codeberg page: <https://codeberg. ... org/gnuhealth/his>

- [mailutils @ Savannah: GNU mailutils version 3.21](#) (2025/12/11 14:31)

GNU mailutils version 3.21 is available for download. Short list of changes in this version follows. Refer to its NEWS file for a detailed discussion. Optionally create intermediate directories when creating mailboxes. New configuration section homedir controls creation of home directories. Imap4d configuration statements create-home-dir and home-dir-mode declared obsolete. Use homedir section, instead. Changes in sieve language Fileinto action: new option :interdir New commands in string expansions: localuser and detail. Immediate values allowed in ldap.field_map configuration statement. Bugfixes Sieve: fix coredump on parsing fileinto :permission action. Sieve: fix optimizer. Library: fix parsing ls-compatible permission strings. Library: fix mu_sieve_machine_clone function. Configuration: use backslash to escape delimiters in some statements. Library: fix localized help output.

- [radius @ Savannah: GNU radius version 1.7](#) (2025/12/11 12:07)

Version 1.7 of GNU radius is available for download. This is a maintenance release. Noteworthy changes: Support for Guile 3.x. Code cleanup. Improved testsuite. Improved documentation. Various bugfixes.

- [FSF News: Free Software Awards winners announced: Andy Wingo, Alx Sa, Govdirectory](#) (2025/12/09 16:40)

BOSTON, Massachusetts, USA (Tuesday, December 9, 2025) — The Free Software Foundation (FSF) announced today the recipients of the 2024 Free Software Awards, which are given annually by the FSF to groups and individuals in the free software community who have made significant contributions to the cause for software freedom.

- [Amin Bandali: Reading and writing emails in GNU Emacs with Gnus](#) (2025/12/06 15:50)

At the 10th anniversary of my involvement in EmacsConf, I'm finally giving my first ever talk at the conference, for EmacsConf 2025. :) In this talk, I give a quick introduction to Gnus and show a basic configuration for reading and writing email with Gnus and Message. You can watch the video below, or from the talk's page on the EmacsConf 2025 wiki: <https://emacsconf.org/2025/talks/gnus> Sorry, this embedded video will not work, because your web browser does not support HTML5 video. [please watch the video in your favourite streaming media player] The above video is provided with closed captions and a transcript — thanks, Sacha! A commented copy of the init file from the video is provided below.

Happy hacking! ;; emacsconf-2025-gnus.el -*- lexical-binding: t -*- ;; This file is marked with CC0 1.0 Universal ;; and is dedicated to the public domain. ;; Note: this file uses the `setopt' macro introduced in Emacs 29 ;; to customize the value of user options. If you are using older ;;

Emacsen, you may can use `customize-set-variable' or `setq'. ;;; Init / convenience ;; Initialize the package system. (require 'package) (package-initialize) (setopt ;; Explicitly set `package-archives', in part to ensure https ones ;; are used, and also to have NonGNU ELPA on older Emacsen as well. package-archives '(("gnu" . "https://elpa.gnu.org/packages/") ("nongnu" . "https://elpa.nongnu.org/nongnu/")) ;; Download descriptions of available packages from the above ;; package archives. (unless package-archive-contents (package-refresh-contents)) ;; Install the keycast package if not already installed. (dolist (package '(keycast)) (unless (package-installed-p package) (package-install package))) ;; Enable keycast to show the current command and its binding in ;; the mode line, for the presentation. (setopt keycast-mode-line-remove-tail-elements nil) (when (fboundp #'keycast-mode-line-mode) (keycast-mode-line-mode 1)) ;; Set a font with larger size for the presentation. ;; It requires that the Source Code Pro be installed on your ;; system. Feel free to comment out or remove. (when (display-graphic-p) (with-eval-after-load 'faces (let ((f "Source Code Pro Medium-15")) (set-face-attribute 'default nil :font f) (set-face-attribute 'fixed-pitch nil :font f)))) ;; Inline function for expanding file and directory names inside ;; `user-emacs-directory'. For example: (+emacs.d "gnus/") (defsubst +emacs.d (path) "Expand PATH relative to `user-emacs-directory'." (expand-file-name (convert-standard-filename path) user-emacs-directory)) (keymap-global-set "C-c e e" #'eval-last-sexp) ;; Add the info directory from the GNU Emacs source repository to ;; the list of directories to search for Info documentation files. ;; Useful if you're using Emacs directly built from a source ;; repository, rather than installed on your system. (with-eval-after-load 'info (setq Info-directory-list `(@Info-directory-list ,(expand-file-name (convert-standard-filename "info/") source-directory) "/usr/share/info/"))) [] ;;; Gnus configuration ;; (info "(gnus) Don't Panic") (keymap-global-set "C-c g" #'gnus) (setopt user-full-name "Gnus Fan Emacsian" user-mail-address "ec25gnus@kelar.org") ;; Tell Emacs we'd like to use Gnus and its Message integration ;; for reading and writing mail. (setopt mail-user-agent 'gnus-user-agent read-mail-command #'gnus) ;; Consolidate various Gnus files inside a gnus directory in the ;; `user-emacs-directory'. (setopt gnus-home-directory (+emacs.d "gnus/") gnus-directory (+emacs.d "gnus/news/") message-directory (+emacs.d "gnus/mail/") nndraft-directory (+emacs.d "gnus/drafts/")) (setopt ; don't bother with .newsrc, use .newsrc.eld instead gnus-save-newsrc-file nil gnus-read-newsrc-file nil) ;; Don't prompt for confirmation when exiting Gnus. (setopt gnus-interactive-exit nil) ;; Configure two IMAP mail accounts. (setopt gnus-select-method '(nnil "") gnus-secondary-select-methods '((nnimap "ec25gnus" (nnimap-stream tls) (nnimap-address "mail.kelar.org") ;; (nnimap-server-port 993) ; imaps (nnimap-authenticator plain) (nnimap-user "ec25gnus@kelar.org")) (nnimap "ec25work" (nnimap-stream tls) (nnimap-address "mail.kelar.org") ;; (nnimap-server-port 993) ; imaps (nnimap-authenticator plain) (nnimap-user "ec25work@kelar.org")) ;; Archive messages into yearly Archive folders upon pressing ;; 'E' (for Expire) in the summary buffer. (nnmail-expiry-wait immediate) (nnmail-expiry-target nnmail-fancy-expiry-target) (nnmail-fancy-expiry-targets (('("from" ".*" "nnimap+ec25work:Archive.%Y"))))) ;; `init-file-debug' corresponds to launching emacs with --debug-init (setq nnimap-record-commands init-file-debug) ;; The "Sent" folder (setopt gnus-message-archive-group "nnimap+ec25gnus:INBOX") ;;; Group buffer ;; Always show INBOX groups even if they have no unread or ticked ;; messages. (setopt gnus-permanently-visible-groups ":INBOX\$") ;; Enable topic mode in the group buffer, for classifying groups. (add-hook 'gnus-group-mode-hook #'gnus-topic-mode) ;;; Article buffer ;; Display the following message headers in Article buffers, ;; in the given order. (setopt gnus-sorted-header-list '("^From:" "^X-RT-Originator" "^Newsgroups:" "^Subject:" "^Date:" "^Envelope-To:" "^Followup-To:" "^Reply-To:" "^Organization:" "^Summary:" "^Abstract:" "^Keywords:" "^To:" "^[BGF]?Cc:" "^Posted-To:" "^Mail-Copies-To:" "^Mail-Followup-To:" "^Apparently-To:" "^Resent-From:" "^User-Agent:" "^X-detected-operating-system:" "^X-Spam_action:" "^X-Spam_bar:" "^Message-ID:" ;; "^References:" "^List-Id:" "^Gnus-Warning:")) ;;; Summary buffer ;; Fine-tune sorting of threads in the summary buffer. ;; See: (info "(gnus) Sorting the Summary Buffer") (setopt gnus-thread-sort-functions '(gnus-thread-sort-by-number gnus-thread-sort-by-subject gnus-thread-sort-by-date)) ;;; Message and

```
sending mail (setopt ;; Automatically mark Gcc (sent) messages as read. gnus-gcc-mark-as-read t ;; Configure posting styles for per-account Gcc
groups, and SMTP ;; server for sending mail. See: (info "(gnus) Posting Styles") ;; Also see sample .authinfo file provided below. gnus-posting-
styles '(("nimap\+ec25gnus:.*" (address "ec25gnus@kelar.org") ("X-Message-SMTP-Method" "smtp mail.kelar.org 587") (gcc
"nimap+ec25gnus:INBOX")) ("nimap\+ec25work:.*" (address "ec25work@kelar.org") ("X-Message-SMTP-Method" "smtp dasht.kelar.org 587")
(gcc "nimap+ec25work:INBOX")))) (setopt ;; Ask for confirmation when sending a message. message-confirm-send t ;; Wrap messages at 70
characters when pressing M-q or when ;; auto-fill-mode is enabled. message-fill-column 70 ;; Forward messages (C-c C-f) as a proper MIME part.
message-forward-as-mime t ;; Send mail using Emacs's built-in smtpmail library. message-send-mail-function #'smtpmail-send-it ;; Omit our own
email address(es) when composing replies. message-dont-reply-to-names "ec25\\(gnus\\|work\\)@kelar\\.org" gnus-ignored-from-addresses
message-dont-reply-to-names) ;; Unbind C-c C-s for sending mail; too easy to accidentally hit ;; instead of C-c C-d (save draft for later) (keymap-
set message-mode-map "C-c C-s" nil) ;; Display a `fill-column' indicator in Message mode. (add-hook 'message-mode-hook #'display-fill-column-
indicator-mode) ;; Enable Flyspell for on-the-fly spell checking. (add-hook 'message-mode-hook #'flyspell-mode) Sample ~/.authinfo file: machine
ec25gnus login ec25gnus@kelar.org password hunter2 machine ec25work login ec25work@kelar.org password badpass123 machine
mail.kelar.org login ec25gnus@kelar.org password hunter2 machine dasht.kelar.org login ec25work@kelar.org password badpass123 Note that
for purpose of storing credentials for use by Gnus's select methods, the machine portions need to match the names we give our select methods
when configuring gnus-secondary-select-methods, namely ec25gnus and ec25work in our example. We also store a copy of the credentials for
use by Emacs's smtpmail when sending mail, where the machine must be the fully-qualified domain name (FQDN) of the SMTP server we specify
with the X-Message-SMTP-Method header for each account by defining a corresponding rule for it in gnus-posting-styles. Lastly, I recommend
using an encrypted authinfo file by saving it as ~/.authinfo.gpg instead to avoid storing your credentials in plain text. If you set up Emacs's
EasyPG, it will seamlessly decrypt or encrypt the file using GPG when reading from or writing to it. Type C-h v auth-sources RET to see the
documentation of the auth-sources variable for more details.
```

- [GNU Guix: A Planet for Guix](#) (2025/12/05 12:00)

I am pleased to announce the availability of Planet Guix, an Atom and RSS aggregator covering all things Guix. You can browse posts on the website or use your favourite feed reader to subscribe to the aggregate feed. Planet Guix already has subscriptions to 19 blogs from around the community; if you write about Guix (no matter how infrequently) and would like your blog to be included, or if you would like to suggest another blog I missed, please create a pull request against the repository in Codeberg — you'll see that the subscriptions are simply configured as association lists in planet/config.scm. Background Back in September, Sébastien Gendre asked on the help-guix mailing list if there were any plans to create a Planet website for Guix. The discussion drifted into how this might be implemented in Guile, and I thought it sounded like an interesting project for the dark autumn evenings. The original Planet aggregator was written in Python and many Planet websites are still using its successor, Venus. The Venus code base has not seen much activity in the last decade and still uses Python 2, which was sunset in 2020. This was all the incentive I needed to implement a new Planet aggregator and static site generator in Guile. Implementation We already know from the likes of Haunt that Guile has all the tools needed to generate a static web site. Both Atom and RSS are XML formats, and Guile also has great support for working with XML. The Guile Planet implementation uses the following built-in modules: (web client) to fetch the feeds. (sxml simple) for reading the Atom/RSS feeds and writing the aggregate Atom feed. (sxml xpath) for searching the feeds to extract the data of interest. (sxml transform) for sanitizing HTML in the entry summaries. Many feeds include HTML content in the entry summary, which we need to parse. This is

where `htmlprag` from `guile-lib` comes in. I used this both to parse HTML embedded in feeds and to generate the static content from an SXML data structure. With these libraries to hand the code for the planet aggregator almost wrote itself! I was trying to keep dependencies to a minimum, but `guile-filesystem` is too useful to do without and, later in the development process, I pulled in `guile-srfi-235` which provides some useful combinators. At the moment I'm only using `apply-chain` to build a function for post-processing one of the feeds, but why re-invent the wheel? Deployment I initially deployed the Planet to a test site running on one of my servers, but the idea was received enthusiastically by the Guix maintainers and I was happy that they wanted to host it on their infrastructure. Of course they are using Guix to manage their virtual machines in Hetzner cloud! While they could have picked up the Planet code and run with it, instead they pointed me at the server configuration and invited me to make a pull request against `hydra/guix-hetzner-2.scm`. They suggested I base the configuration on their existing `static-web-site-configuration` so I started reading the code which proved very educational (I admit that I had to sleep on it for a week before coming up with a plan!) The `static-web-site-configuration` did almost everything needed to build the Planet aggregator, only the build step runs like a Guix package build in an isolated environment with no network - so we cannot fetch the feeds in this build step. Luckily, I had already implemented functionality in the Planet code base to build the static site from feeds cached on disk. So it was simply a case of adding support for a pre-build script to the `static-web-site-configuration` and using this step to download the feeds. The pull request was merged after some short discussion, and a few days later the site was live in its new home. Community This was my second time contributing to the Guix project and I'm pleased to report that it was a smooth experience both times. When it came to the deployment, I was glad that I was encouraged to add the service configuration myself instead of being spoon-fed: working with computers, you learn best by doing. I'd like to give a shout-out to `@civodul`, `@cbaines`, and `@apteryx` for their help with the deployment, and to the several people who sent merge requests to add their blogs before I even got around to writing this announcement. I think the Planet site is already a great place to discover people writing about Guix, and I hope it grows and becomes an asset to the community. Happy reading!

- [a2ps @ Savannah: a2ps 4.15.8 released \[stable\]](#) (2025/12/04 17:34)

I am delighted to announce a new release of GNU a2ps, the “anything to PostScript” system. This is to announce a2ps-4.15.8, a stable release. This release fixes a buffer overflow, and a failure to build on some older systems. There have been 13 commits by 1 people in the 21 weeks since 4.15.7. See the NEWS below for a brief summary. Thanks to everyone who has contributed! The following people contributed changes to this release: Reuben Thomas (13) Reuben [on behalf of the a2ps maintainers]

==== Here is the GNU a2ps home page:
<https://gnu.org/s/a2ps/> Here are the compressed sources and a GPG detached signature: <https://ftpmirror.gnu.org/a2ps/a2ps-4.15.8.tar.gz>
<https://ftpmirror.gnu.org/a2ps/a2ps-4.15.8.tar.gz.sig> Use a mirror for higher download bandwidth: <https://www.gnu.org/order/ftp.html> Here are the SHA256 and SHA3-256 checksums: File: a2ps-4.15.8.tar.gz SHA256 sum:
8d13915a36ebbfa8e7b236b350cc81adc714acb217a18e8d8c60747c0ad353f9 SHA3-256 sum:
0dce19c25df3be0ce1fc2b92710c33a724595b3617686cbb904ab60dcdd15b34 Verify the SHA256 checksum with either `sha256sum`, `sha256`, or `shasum -a 256`. Verify the SHA3-256 checksum with `cksum -a sha3 --check` from `coreutils-9.8`. Use a .sig file to verify that the corresponding file (without the .sig suffix) is intact. First, be sure to download both the .sig file and the corresponding tarball. Then, run a command like this: `gpg --verify a2ps-4.15.8.tar.gz.sig` The signature should match the fingerprint of the following key: `pub rsa2048 2013-12-11 [SC] 2409 3F01 6FFE 8602 EF44 9BB8 4C8E F3DA 3FD3 7230 uid Reuben Thomas <rrt@sc3d.org> uid keybase.io/rrt <rrt@keybase.io>` If that command

fails because you don't have the required public key, or that public key has expired, try the following commands to retrieve or refresh it, and then rerun the 'gpg --verify' command. `gpg --locate-external-key rrt@sc3d.org gpg --recv-keys 4C8EF3DA3FD37230 wget -q -O- 'https://savannah.gnu.org/project/release-gpgkeys.php?group=a2ps&download=1' | gpg --import -` As a last resort to find the key, you can try the official GNU keyring: `wget -q https://ftp.gnu.org/gnu/gnu-keyring.gpg gpg --keyring gnu-keyring.gpg --verify a2ps-4.15.8.tar.gz.sig` This release is based on the a2ps git repository, available as `git clone https://https.git.savannah.gnu.org/git/a2ps.git` with commit `c41304d1745017322adb12ba40e7f06984e7f14e` tagged as v4.15.8. For a summary of changes and contributors, see: <https://gitweb.git.savannah.gnu.org/gitweb/?p=a2ps.git;a=shortlog;h=v4.15.8> or run this command from a git-cloned a2ps directory: `git shortlog v4.15.7..v4.15.8` This release was bootstrapped with the following tools: Autoconf 2.72 Automake 1.18.1 Gnulib 2025-12-04 481064c5c22c8137188eecb6662ebeb03fc6d0b8 NEWS * Noteworthy changes in release 4.15.8 (2025-12-04) [stable] * Bug fixes: - Fix a buffer overflow when a long value supplied to -E. - Include some header files with system paths, not user paths. * Build system: - Fix building on systems that need gnulib's malloc wrapper. - Remove a generated file from git. - Update the version of gettext used. * Documentation: - Update copyright notices to point to GPL online.

- [GNU Guile: GNU Guile 3.0.11 released](#) (2025/12/01 14:00)

We are pleased to announce the release of GNU Guile 3.0.11! This release is mainly a bug-fix release, though it does include a number of new features, including support for SRFI 197: Pipeline Operators, support for SRFI 207: String-notated bytevectors (bytestrings), and JIT (just-in-time) compilation for the RISC-V architecture. It also overhauls SRFI-64 (testing) and includes many other improvements. For full details, see the release notes and check out the download page. Happy Guile hacking!

- [Amin Bandali: Free software activities in November 2025](#) (2025/11/30 23:26)

Hello and welcome to my November free software activities report. I've been working on a number of things throughout this month but they're not quite ready for reporting yet, so this month's report will be quite short. GNU & FSF EmacsConf: I recorded the video for my Gnus talk for this year's conference. The video will be available along with the other EmacsConf talks from the conference website, but if you're feeling particularly impatient you can sneak a peek at it. :) <https://archive.org/details/emacsconf-2025-gnus> GNU Spotlight: I prepared and sent the November GNU Spotlight to the FSF campaigns team for publication on the FSF's community blog and the monthly Free Software Supporter newsletter. Take care, and so long for now.

- [Jose E. Marchesi: Algol 68 Front-End merged in GCC](#) (2025/11/30 00:00)

I am very happy to announce that, after almost a year in the works, today the Algol 68 front-end has been merged in GCC proper in its development trunk branch. This means that we are no longer off-tree, and that GCC 16 will be featuring a full-fledged and modern Algol 68 compiler once it gets released. An Algol 68 module This doesn't mean the work is done. The modules system, which is already functional, needs to be completed and polished, parallel clauses are still to be implemented, and the quality of the generated code should be generally improved. We also want to replace the Boehm GC with a tightly integrated customized, exact one, and to continue modernizing and expanding the language, always carefully and respectfully, via GNU extensions: exception handling, FFI, etc. At this point I would like to thank Marcel van der Veer, Pietro Monteiro, Mohammad-Reza Nabipoor, Thomas Schwinge, Sam James, Matthias Klose, Iain Buclaw, Andrew Pinski, Segher Boessenkool, Iain Sandoe, the GCC global reviewers and the overall GCC community. Without their help, and Marcel's nifty Algol 68 parser, this front-end would simply not exist. And now, the real fun starts... ;) Happy algoling!

- [Simon Josefsson: Container Images for Debian with Guix \(2025/11/28 16:32\)](#)

The debian-with-guix-container project build and publish container images of Debian GNU/Linux stable with GNU Guix installed. The images are like normal Debian stable containers but have the guix tool and a reasonable fresh guix pull. Supported architectures include amd64 and arm64. The multi-arch container is called: `registry.gitlab.com/debdistutils/guix/debian-with-guix-container:stable` It may also be accessed via `debian-with-guix` at Docker Hub as: `docker.io/jas4711/debian-with-guix:stable` The container images may be used like this: `$ podman run --privileged -it --hostname guix --rm registry.gitlab.com/debdistutils/guix/debian-with-guix-container:stable root@guix:/# hello bash: hello: command not found root@guix:/# guix describe guix c9eb69d repository URL: https://gitlab.com/debdistutils/guix/mirror.git branch: master commit: c9eb69ddb05e77300b59f49f4bb5aa50cae0892 root@guix:/# LC_ALL=C.UTF-8 /root/.config/guix/current/bin/guix-daemon --build-users-group=guixbuild & [1] 21 root@guix:/# GUIX_PROFILE=/root/.config/guix/current; . "$GUIX_PROFILE/etc/profile" root@guix:/# guix describe Generation 2 Nov 28 2025 10:14:11 (current) guix c9eb69d repository URL: https://gitlab.com/debdistutils/guix/mirror.git branch: master commit: c9eb69ddb05e77300b59f49f4bb5aa50cae0892 root@guix:/# guix install --verbosity=0 hello accepted connection from pid 55, user root The following package will be installed: hello 2.12.2 hint: Consider setting the necessary environment variables by running: GUIX_PROFILE="/root/.guix-profile" . "$GUIX_PROFILE/etc/profile" Alternately, see `guix package --search-paths -p "/root/.guix-profile"`. root@guix:/# GUIX_PROFILE="/root/.guix-profile" root@guix:/# . "$GUIX_PROFILE/etc/profile" root@guix:/# hello Hello, world! root@guix:/# Below is an example GitLab pipeline job that demonstrate how to run guix install to install additional dependencies, and then download and build a package that pick up the installed package from the system. test-wget-configure-make-libksba-amd64: image: registry.gitlab.com/debdistutils/guix/debian-with-guix-container:stable before_script: - env LC_ALL=C.UTF-8 /root/.config/guix/current/bin/guix-daemon --build-users-group=guixbuild $GUIX_DAEMON_ARG & - GUIX_PROFILE=/root/.config/guix/current; . "$GUIX_PROFILE/etc/profile" - guix describe - guix install libgpg-error - GUIX_PROFILE="/root/.guix-profile"; . "$GUIX_PROFILE/etc/profile" - apt-get install --update -y --no-install-recommends build-essential wget ca-certificates bzip2 script: - wget https://www.gnupg.org/ftp/gcrypt/libksba/libksba-1.6.7.tar.bz2 - tar xfa libksba-1.6.7.tar.bz2 - cd libksba-1.6.7 - ./configure - make V=1 - make check VERBOSE=t V=1 The images were initially created for use in GitLab CI/CD Pipelines but should work for any use. The images are built in a GitLab CI/CD pipeline, see .gitlab-ci.yml. The containers are derived from official Debian stable images with Guix installed and a successful run of guix pull, built using buildah invoked from build.sh using image/Containerfile that runs image/setup.sh. The pipeline also push images to the GitLab container registry, and then also to Docker Hub. Guix binaries are downloaded from the Guix binary tarballs project because of upstream download site availability and bandwidth concerns. Enjoy these images! Hopefully they can help you overcome the loss of Guix in Debian which made it a mere apt-get install guix away before. There are several things that may be improved further. An alternative to using podman --privileged is to use --security-opt seccomp=unconfined --cap-add=CAP_SYS_ADMIN,CAP_NET_ADMIN which may be slightly more fine-grained. For ppc64el support I ran into an error message that I wasn't able to resolve: guix pull: error: while setting up the build environment: cannot set host name: Operation not permitted For riscv64, I can't even find a Guix riscv64 binary tarball for download, is there one anywhere? For arm64 containers, it seems that you need to start guix-daemon with --disable-chroot to get something to work, at least on GitLab.com's shared runners, otherwise you will get this error message: guix install: error: clone: Invalid argument Building the images themselves also require disabling some security functionality, and I was not able to build images with buildah without providing --cap-add=CAP_SYS_ADMIN,CAP_NET_ADMIN otherwise there were errors like this: guix pull: error: cloning builder process: Operation not permitted guix pull: error: clone: Operation not permitted guix pull: error: while setting up the build environment: cannot`

set loopback interface flags: Operation not permitted Finally on amd64 it seems --security-opt seccomp=unconfined is necessary, otherwise there is an error message like this, even if you use --disable-chroot: guix pull: error: while setting up the child process: in phase setPersonality: cannot set personality: Function not implemented This particular error is discussed upstream, but I think generally that these error suggest that guix-daemon could use more optional use of features: if some particular feature is not available, gracefully fall back to another mode of operation, instead of exiting with an error. Of course, it should never fall back to an insecure mode of operation, unless the user requests that. Happy Hacking!

- [gnuastro @ Savannah: Gnuastro 0.24 released](#) (2025/11/27 01:35)

The 24th release of GNU Astronomy Utilities (Gnuastro) is now available. See the full announcement for all the new features in this release and the many bugs that have been found and fixed: <https://lists.gnu.org/archive/html/planet.gnu/2025-11/msg00001.html>

From:

<https://wiki.tromjaro.alexio.tf/> - **TROMjaro wiki**

Permanent link:

<https://wiki.tromjaro.alexio.tf/doku.php?id=news:planet:gnu&rev=1583652632>

Last update: **2021/10/30 11:38**

