GNU Planet - Latest News

- FSF Events: Free Software Directory meeting on IRC: Friday, November 6, starting at 12:00 EST (17:00 UTC) (2026/11/03 21:35)
  Join the FSF and friends on Friday, November 6 from 12:00 to 15:00 EST (17:00 to 20:00 UTC) to help improve the Free Software Directory.
- FSF Blogs: October GNU Spotlight with Amin Bandali featuring fifteen new GNU releases: Hyperbole, Gxmessage, and more! (2026/11/02 19:13)
- automake @ Savannah: automake-1.19.92 released [beta] (2026/10/31 16:39)
  This is to announce automake-1.19.92, a alpha release. It is the first (and hopefully final) pre-release for the upcoming Automake 1.20. Please consider doing us a favour and testing it on different platforms. This release brings important bug-fixes that resolve some long-standing limitations and adds minor new features to automake. There have been 80 commits by 3 people in the 49 days since 1.19. See the NEWS below for a brief summary. Thanks to everyone who has contributed! The following people contributed changes to this release: Kamila Szewczyk (75) Karl Berry (4) Paul Eggert (1) Kamila Szewczyk (szewczyk@math.uni-sb.de) [on behalf of the automake maintainers] ========== Download here: https://alpha.gnu ... ke-1.19.92.tar.gz https://alpha.gnu ... ke-1.19.92.tar.xz Please report bugs and problems to <bug-automake@gnu.org> (instead of replying to this mail), and send general comments and feedback to <automake@gnu.org>, and patches to <automake-patches@gnu.org>. Thanks to everyone who has reported problems, contributed patches, and helped test Automake! -*-*-*- If you install this test release in its own prefix (recommended) and you use libtool, you'll need to arrange for the libtool m4 files to be found by aclocal. For info on this, see: https://gnu.org/s ... TOOL-is-undefined For planned incompatibilities in a possible future Automake 2.0 release, please see NEWS-future and start following the advice there now. ~~~~~~~~~~ New in 1.19.2 (2026-10-28): New features added - New configure option --enable-jobserver lets compilers and linkers inherit GNU make's pipe jobserver, for example for GCC's -flto=jobserver. Dry-run, question, and touch modes still do not run these commands. (bug #45756) - New option delete-on-error. It emits the .DELETE_ON_ERROR special target in the generated Makefile.in, instructing GNU make to delete the target when its recipe fails after having modified it. A failed rule like `foo.h: gen.sh; ./gen.sh > $@' then no longer leaves a truncated result that later runs of make treat as up to date. Other make programs do not support this, so the option will have no effect. - New target print-distfiles lists the files that "make dist" would distribute. Changes to Automake-generated Makefiles - distcheck no longer runs "make dvi" by default, so TeX is not needed to check a distribution containing Texinfo manuals. Set AM_DISTCHECK_DVI_TARGET in Makefile.am to, e.g., dvi, pdf or html, to check that the documentation can be built. (bug #79053) Bugs fixed - Repeated target declarations in overlapping Automake conditionals no longer produce spurious redefinition warnings when adding prerequisites. (bug #13940) - Dependencies on another target's objects in LIBADD or LDADD are now tracked regardless of the order in which targets are processed. (bug #60508) - distcheck now detects symbolic links left in the installation tree after uninstall, including dangling links. - lex and yacc output files distributed from a VPATH build no longer carry "#line" directives pointing to their sources through the build-time srcdir, e.g. "../src/parse.y"; the distributed copies now refer to the sources next to them. (bug #39666) - dist-zstd now honors numeric ZSTD_CLEVEL values, such as 3, instead of treating them as input filenames. This fixes failed or incorrect distribution archives when ZSTD_CLEVEL is set. - lex and yacc

rules now create output directories before generating sources, fixing out-of-tree builds with subdir-objects and --disable-dependency-tracking. (bug #27188, Debian bug #807050) - configure now refuses a source directory path containing '<', '>', '(' or ')'. These could be interpreted as shell redirections or syntax in compiler commands, potentially overwriting files in the build directory. - configure now refuses a source directory path ($srcdir) containing ';' or '|', reporting "unsafe srcdir value", as it already did for characters such as '$' and '&'. Such a path used to fail later with misleading errors, from config.status's dependency-tracking setup or from make ("No rule to make target"). These characters remain accepted when $srcdir is '.'. - Non-recursive Vala builds now pass distcheck when generated Vala sources trigger recompilation in the build directory. Generated C files are moved into their subdirectories together, and build-tree stamp files are removed by distclean. (bug #76527) ~~~~~~ Here is the GNU automake home page: https://gnu. ... /s/automake/ Here are the compressed sources: https://alpha.gnu ... ke-1.19.92.tar.gz (2.5MB) https://alpha.gnu ... ke-1.19.92.tar.xz (1.7MB) Here are the GPG detached signatures: https://alpha.gnu ... .19.92.tar.gz.sig https://alpha.gnu ... .19.92.tar.xz.sig Use a mirror for higher download bandwidth: https://www.gnu.o ... rg/order/ftp.html Here are the SHA256 and SHA3-256 checksums: File: automake-1.19.92.tar.gz SHA256 sum: 8f766f5f78d98f81ce30446cab79caf2d9b514f32821e2c466c846a22f971b68 SHA3-256 sum: 4445cde757b16c43ec389461d2bab9f99562586ab07a4a0cf506e3450e1eb576 File: automake-1.19.92.tar.xz SHA256 sum: 3e338607daf027546f2040439578176feab6ee8efab83dc9a9a9680884ff1fb5 SHA3-256 sum: 84348e4b626558f1aacecde044400131fbdcbbde74fe447a338427b580d4174d Verify the SHA256 checksum with either sha256sum, sha256, or 'shasum -a 256'. Verify the SHA3-256 checksum with 'cksum -a sha3 -l 256' 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 automake-1.19.92.tar.gz.sig automake-1.19.92.tar.gz The signature should match the fingerprint of the following key: pub rsa4096 2022-07-31 [C] 6C22 2EA6 B2BD 216A A406 516A C868 F0B6 DE38 409D uid Kamila Szewczyk <k@iczelia.net> uid Kamila Szewczyk <szewczyk@cs.uni-saarland.de> uid Kamila Szewczyk <iczelia@gnu.org> uid Kamila Szewczyk <szewczyk@math.uni-sb.de> 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 --recv-keys 6C222EA6B2BD216AA406516AC868F0B6DE38409D As a last resort to find the key, you can try the official GNU keyring: wget -q https://ftp.gnu.o ... u/gnu-keyring.gpg gpg --keyring gnu-keyring.gpg --verify automake-1.19.92.tar.gz.sig automake-1.19.92.tar.gz This release is based on the automake git repository, available as git clone https://https.git ... /git/automake.git with commit f6104d007b721055edec9c689fdc9c397623600a tagged as v1.19.92. For a summary of changes and contributors, see: https://gitweb.gi ... ortlog;h=v1.19.92 or run this command from a git-cloned automake directory: git shortlog v1.19..v1.19.92 -- With Valediction, Kamila Szewczyk (https:// ... elia.net)

- FSF Blogs: Bruce Schneier on why encryption is important (2026/10/30 21:25)
- parallel @ Savannah: GNU Parallel 20261022 ('Lycée') released [stable] (2026/10/28 20:56)
  GNU Parallel 20261022 ('Lycée') has been released. It is available for download at: lbry://@GnuParallel:4 Quote of the month: gnu parallel is amazing -- Tarun Agarwal @axrawal New in this release: Bug fixes and man page updates. 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 ... rg/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.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/1 ... 81/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.

- gxmessage @ Savannah: gxmessage 4.1.1 for GTK4 released (2026/10/25 01:51)
  gxmessage is a GTK based alternative to xmessage. Version 4.1.0 is a long overdue initial port from GTK3 to GTK4. It also includes a new search feature (Ctrl-F). Version 4.1.1 is a post-port tidy-up. It adds a new desktop icon and improves automatic window sizing.
- Amin Bandali: Living healthier (2026/10/24 21:01)
  Earlier this week I had a coaching session with Protesilaos, where we focused on health and wellness. I took some notes during our call, which I

expand on here, and share with Prot's permission. I've been paying more serious attention to my health this year, and our discussion was in the context of me striving to live a healthier life. These notes are shared in the hope of being useful and are not professional advice; for medical advice please consult a licensed physician. Update 24 October 2026: Prot published a note about our discussion on his website: Healthier lifestyle with Amin Bandali Exercise Life in today's fast-paced world can feel busier than ever, where every day is packed with a myriad of activities and tasks that need to be done, or tempting distractions that pop up seemingly randomly, with hardly a moment to spare. Despite that, or perhaps because of that, it's worthwhile to make time for ourselves and the various aspects of our health. As embodied souls, our bodies condition our minds. We take care of our body so we can take care of our mind. As humans we evolved for an active lifestyle out of survival necessity. In today's society many of us have sedentary office jobs that involve sitting for long periods of time. It's especially important that we try and make time for some regular physical activity like exercising and resistance training, to help build and retain muscle mass and mobility as we age. Fitness does not have to be the end goal that takes over our life, though there is nothing wrong with that if fitness is one's central goal e.g. as a personal trainer. Rather, for someone like me with other primary goals, my reason to exercise is to help live a healthier life, so I could spend quality time with my loved ones and carry out my responsibilities with a healthy body and mind. Prot mentioned that our health is directly dependent on the following four factors: what we eat; what we drink; what we breathe; and what we think. Thus compromising any of these - for example, eating greasy food or drinking alcohol - would undermine the gains from exercising. While pursuing fitness, it's important to have sustained effort so that the body remains active and the mind is not distracted, and equally important to have sustainable effort so that we don't suffer injuries and not make the pursuit of fitness burdensome financially. Calisthenics is a key part of the strength training activities Prot practices, in addition to the physically intensive work he does on his land. Calisthenics [...] is a form of strength training that utilizes an individual's body weight as resistance to perform multi-joint, compound movements with little or no equipment. Calisthenics are popular for their convenience, as they require little to no equipment and can be performed anywhere. — Calisthenics, Wikipedia Calisthenics can be done progressively, starting small and building in intensity and duration over time. It is also economically/financially sustainable. These calisthenic exercises are a good starting point (barring existing injuries in the related areas): push-ups - easier version: at an angle or against the wall; pull-ups - easier version: at an angle e.g. using door frame; squats - easier version: don't squat all the way down; and sit-ups. Aim to perform each exercise in 3 sets of 10 repetitions, but it's perfectly okay to start with fewer and gradually build up. The exercises should be done around the same time everyday, so that there is enough time for the muscles to rest and recover for the next day. Doing them at night then again the next morning, for instance, would be too short of a time period and not enough for the body to recover. It's also vital to get enough sleep each night to make sure the body can properly rest and be energized the next day. Food As mentioned earlier, it's crucial to pair exercise with a healthy diet. I found Prot's programming metaphor for food interesting: similar to how it takes more time and effort for us to process and understand complex source code, breaking down and processing complex foods is similarly more difficult for the body. Prot does not eat fried food, and recommends avoiding sauces as much as possible: greasy, salty, and creamy extras make food more irresistible. They tend to make one want to eat more than what the body needs, and are to be avoided in isolation but especially in combination. In keeping ingredients close to their natural states, not overdoing things, and not eating more than the body needs, Prot extends his philosophy of simplicity and moderation to cooking. In addition to eating healthy meals, we need to apply similar consideration to snacks: avoiding overly salty or sugary snacks, and choosing healthier ones. Nuts can make great snacks in small quantities, especially almonds and walnuts, if one's not allergic. In addition to the inherent healthiness of the food or snack, Prot also suggests paying attention to methods of preparation. For instance, regarding concentration or distillation: juices,

even when freshly squeezed from fruits, might have more of the fruits' sugar than its pulp and other beneficial parts. Additionally, chewing triggers biological responses in our body which for example can reduce stress, and release satiety hormones to help avoid overeating. So, Prot suggests to just eat the fruits themselves. Regarding weight loss, it will occur naturally as we introduce exercise into our routine, and pay closer attention to what we eat and drink. It's important to lose the weight slowly to condition our body to the new reality as it changes. Shortcuts tend to either not work, or have negative consequences. By going through the process slowly and with regularity, we gradually retrain our muscle memories and adjust our habits so we can maintain the lower weight and the healthier lifestyle. Regularity is key: it's better do a little everyday, than do a lot on only two days. It helps us develop the attitude that the exercise is part of our regular effort that we do consistently all the time. Working on wellness and living healthier is a matter of moderation, and building capacity over time. I will give it the time and attention it needs, without rushing, and enjoy the journey to becoming a healthier version of myself. Take care, and so long for now.

- pspp @ Savannah: PSPP 2.1.3 has been released. (2026/10/23 17:01)
  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 2.1.2 to 2.1.3: Bug fixes. MATRIX: New MEAN and COV extension functions. Please send PSPP bug reports to bug-gnu-pspp@gnu.org.
- FSF Blogs: Election integrity requires free software (2026/10/22 20:55)
- nano @ Savannah: GNU nano 9.3 was released (2026/10/22 09:04)
  Version 6.0 brought option --zero, for hiding all interface elements. Version 7.0 brought braced function names, to make string binds more readable and stable and versatile. Version 8.0 changed Ctrl+F to mean "forward search", like in most other programs, and added option --modernbindings for making most bindings like in other programs. Version 9.0 brought horizontal scrolling of the viewport. Along the way many small issues were fixed and little improvements made.
- GNU Taler news: Taler Operations AG is getting ready to operate Taler in Germany (2026/10/19 22:00)
  This post explains why this company is needed, what changed in the NGI TALER consortium, and how citizens who want to see Taler succeed can help, including as investors.
- libunistring @ Savannah: GNU libunistring-1.5 released (2026/10/17 22:16)
  Download from https://ftp.gnu.o ... string-1.5.tar.gz This is a stable release. New in this release: The data tables and algorithms have been updated to Unicode version 18.0.0.
- www-zh-cn @ Savannah: Welcome our new member - Richard Qin (2026/10/12 06:30)
  Hi, All: Please join me in welcoming our new member: User Details: ---------- Name: Richard Qin Login: richardqin Email: richard.qin.x@gmail.com I wish Richard Qin a wonderful journey in GNU CTT. Happy Hacking wxie
- FSF Blogs: Raspberry Pi puts digital handcuffs on RAM upgrades (2026/10/09 16:30)
- GNU Guix: Extending Guix (2026/10/08 11:00)
  Guix is all about empowering people, so it should come as no surprise that one can extend it with new guix commands. You can show the commands available in your current Guix through the help command:$ guix helpIf you are using a recent Guix, there are a number of extensions at your disposal, they are available as individual packages, and as a meta-package providing a collection of extensions. Try them out with:$ guix shell guix guile guix-extension-collectionI'm adding the guix and guile packages to the shell because we want the different Guile and Guix search

paths to be adjusted in the shell. To know more about why this is needed, read Search Paths.You will notice that the help menu has been extended with information about the available extensions:$ guix help ... extension commands explore interactively explore a Guix System configuration removals keep up to date with package removals compose docker compose compatibility layer toys Explore packages and services through REST API xsearch search for packages using a fast Xapian cache ...Since this blog post is called "Extending Guix", let's write an extension, shall we?How does Guix locate extensions?Guix locates extensions by looking up GUIX_EXTENSIONS_PATH. Recently Guix has introduced a new way of writing extensions. The old way would search extensions under /path/to/guix/extensions and the extensions modules would be named (guix extensions NAME). The new schema expects extensions to be under /path/to/SCHEMA_VERSION; the name of the module will still be (guix extensions NAME).The advantage of this new schema is the Guix can treat the extension module as a standard Guile module, meaning that the runtime is able to find the compiled .go file of the extension. The old schema was relaying on runtime evaluation; the load machinery was not handling compiled files. This has a considerable improvement on performance, so you are encouraged to update any old extension to the new schema.Writing an extensionEnough introductions, let's write a basic extension.The first step is to create the project structure. Remember that the extension machinery expects modules to be named (guix extensions NAME).We start by creating, in our project root, the directories for the extension.$ mkdir -p guix/extensionsNow, we create the file guix/extensions/hello.scm with the following contents:(define-module (guix extensions hello))A blank canvas...We import (guix scripts) to get the define-command macro. We declare it and export it so the extension machinery can find the command in the public interface of the module.(define-module (guix extensions hello) #:use-module (guix scripts) #:export (guix-hello)) (define-command (guix-hello . args) (category extension) (synopsis "say hello") (display "hello, I'm a Guix extension!\n"))With this, we already have a working Guix extension. We can run the extension like this from the root of the project.$ GUIX_EXTENSIONS_PATH=$PWD guix hello hello, I'm a Guix extension!Since we would like our extension to be discoverable by users, we will add a help message which will make the extension display in the guix help menu.We will use (srfi srfi-37) to write the option parser, and (guix ui) for the internationalized strings; you will see that I import these modules when I show you the complete extension. Let's focus on the option definitions:(define (show-help) (display (G_ "Usage: guix hello\n")) (display (G_ "Just say hello, it's not that complicated.\n")) (display (G_ " --help display this message")) (newline)) (define %options (list (option '(#\h "help") #f #f (lambda args (leave-on-EPIPE (show-help)) (exit 0)))))Since we are only handling the arguments for showing the help message, we just need to call the parser at the start of the command:(define-command (guix-hello . args) (category extension) (synopsis "say hello") (parse-command-line args %options (list '())) (display "hello, I'm a Guix extension!\n"))Here you have the complete extension module:(define-module (guix extensions hello) #:use-module (guix scripts) #:use-module (guix ui) #:use-module (srfi srfi-37) #:export (guix-hello)) (define (show-help) (display (G_ "Usage: guix hello\n")) (display (G_ "Just say hello, it's not that complicated.\n")) (display (G_ " --help display this message")) (newline)) (define %options (list (option '(#\h "help") #f #f (lambda args (leave-on-EPIPE (show-help)) (exit 0))))) (define-command (guix-hello . args) (category extension) (synopsis "say hello") (parse-command-line args %options (list '())) (display "hello, I'm a Guix extension!\n"))With that, our extension will be present when we ask Guix for help:$ GUIX_EXTENSIONS_PATH=$PWD guix help ... extension commands hello say helloIt also takes a --help flag:$ GUIX_EXTENSIONS_PATH=$PWD guix hello --help Usage: guix hello Just say hello, it's not that complicated. --help display this messagePackaging a Guix extensionAt the time of writing, we don't have a dedicated Guix build-system for extensions. Fortunately, the guile-build-system is close enough for this use case.Let's make a package definition for our new hello extension that uses our local sources. Create a guix.scm file at the root of the project with the following contents:(use-modules (gnu packages package-management) (guix build-system guile) (guix gexp) (guix git-download) (guix packages)

((guix licenses) #:prefix license:)) (define vcs-file? ;; Return true if the given file is under version control. (or (git-predicate (dirname (canonicalize-path (assq-ref (current-source-location) 'filename)))) (const #t))) (define-public guix-hello (package (name "guix-hello") (version "0.0.0-git") (source (local-file (assume-valid-file-name ".") "pin-checkout" #:recursive? #t #:select? vcs-file?)) (build-system guile-build-system) (arguments (list #:scheme-file-regexp #~(lambda (file stat) (and ((file-name-predicate #$default-scheme-file-regexp) file stat) (not ((file-name-predicate "^(guix|channels|manifest)\\.scm$") file stat)))) #:phases #~(modify-phases %standard-phases (add-after 'build 'move-to-extension-directory (lambda _ (with-directory-excursion #$output (mkdir-p "share/guix/extensions/1.5/guix/extensions") (rename-file (string-append "share/guile/site/" (target-guile-effective-version) "/guix/extensions/hello.scm") "share/guix/extensions/1.5/guix/extensions/hello.scm")))))) (native-inputs (list guix)) (inputs (list (lookup-package-input guix "guile"))) (home-page "https://codeberg.org/guix-extensions") (synopsis "Make Guix say hello") (description "This extension provides the @command{guix hello} command, which makes Guix say hello.") (license license:gpl3+))) guix-helloWe can test this new extension like this:$ guix shell -CW -f guix.scm -- guix helloThe flags passed to guix shell are the following:-C (--container): To prevent your environment from interfering.-W (--nesting): To bring the current Guix you are using into the container so it can load the extension.Since this example is using the new extension scheme, the guix command you are running must be at or newer than commit de069958fc.Closing wordsI hope you find this small introduction to Guix extensions useful and that you start to write your own Guix extensions.I would like to encourage everyone reading this to submit their extensions to the guix-extensions Codeberg organization. The idea is to make this organization a central hub for everyone to participate in the development of useful extensions for the community.

- health @ Savannah: GNU Health as a solution for Europe digital change - Italy Digi-CHange conf (2026/10/03 12:14)
  During the days 6-9 of October will take place the Digi-Change conference in Turin, Piedmont region, Italy. Digi-CHange promotes the digital transformation of medical and nursing education, introducing an inter-professional model that connects educators, clinicians, engineers, and industry innovators. Aligned with the Erasmus+ Capacity Building in Higher Education priority of Digital Transformation, Digi-Change strengthens the capacity of universities in Georgia, Ukraine, and EU partner countries to modernize curricula and foster digital competencies for a sustainable, innovation-ready healthcare workforce. In the first session, Luis Falcón will deliver a keynote presentation about the social impact of GNU Health and Libre/Free Software is having in the public healthcare system, and how it can be used in the context of Telemedicine to provide meaningful and timely healthcare access to remote areas in Europe. Other topics will be building large federated health networks, privacy, and the human factor in healthcare, medical genetics and precision medicine. Dr. Falcon will address the risks of some emerging technology (eg, LLM) poses at different levels (individual, clinical practice, cultural & scientific community and the environment), and ethical alternatives in eHealth and decision support (DSS) During this three-day “train-the-trainers” conference, physicians, nurses and other health professionals from Ukraine, Georgia and Latvia will join colleagues from Italy, France, Austria and other European countries to exchange knowledge and experiences. In the topic of Telemedicine, Alberto Lazzero will address National records and cross-border cooperation among France and Italy. Prof. Dr. Alberto Lazzero is the Head of Telemedicine, Hospital of Briançon (France). He is an specialist in telemedicine for patients in mountain and cross-border areas. The conference will be in Turin, onsite, and will count with experts on diagnostic imaging, cancer and bioinformatics, among others. There will be visits to the Istituto Neurologico “Carlo Besta” (Milan) and INOC (National Oncologic Institute Candiolo). Looking forward to meet our colleagues from Europe in Italy and share the experiences and impact of Free/Libre software in healthcare and our society. Original post: https://my.gnusol ... thumbnail_id=2272
- FSF Blogs: September GNU Spotlight with Amin Bandali featuring seventeen new GNU releases: Coreutils, SCM, and more! (2026/10/01 16:45)

- GNU Taler news: The rise of financial surveillance - Buenos Aires, Argentina (2026/09/27 22:00)
  Christian Grothoff gave a talk on the importance of privacy in digital payment infrastructure and the features of GNU Taler designed for that purpose during his visit to Buenos Aires, Argentina. The event was co-organized by the Argentine Observatory of Information Technology Law (ODIA) and NUMIS, led by Sebastian Marchano.
- gengetopt @ Savannah: 2.23.1 is out (2026/09/27 13:22)
  Various small bug fixes, nothing major.
- GNU Taler news: Taler and the successful attacks against 1024-bit RSA signatures (2026/09/26 22:00)
  Any real and recommended deployment of GNU Taler is not at risk from this attack, due to the following security measures in GNU Taler:
- parted @ Savannah: parted-3.8 release [stable] (2026/09/23 23:28)
  This is to announce parted-3.8, a stable release. This is the same as the 3.7.14 alpha release except that gnulib has been updated. The full set of changes since parted 3.7: Brian C. Lane (21): maint: post-release administrivia README-release: Fix typos in doxygen instructions Cleanup zero as null pointer constant warnings libparted: Catch FAT metadata triggered errors resize: Make sure 32bit build cannot overflow frag_count resize: Make sure hfsc_new_cachetable cannot overflow on 32bit fdasd: Make sure data set name is positive parted: Fix partition number allocation in do_print maint: Update to latest gnulib and bootstrap script maint: Update copyright statements to 2026 NEWS: Update news for next release version 3.7.13 maint: post-release administrivia hurd: Fix gnu_read problems with block size > 512b NEWS: Update news for next release maint: Update to latest gnulib version 3.7.14 maint: post-release administrivia maint: Update to latest gnulib NEWS: Releasing stable version 3.8 version 3.8 Colin Watson (1): maint: Distribute Doxyfile Victor Couty (1): Adding support for ExFAT filesystem bug-parted@gnu.org (1): bug #80795: [PATCH] build: mark functions with const attribute, per gcc warnings Brian [on behalf of the parted maintainers] ========== Here is the GNU parted home page: https://gnu ... g/s/parted/ Here are the compressed sources and a GPG detached signature: https://ftp.gnu.o ... parted-3.8.tar.xz https://ftp.gnu.o ... ed-3.8.tar.xz.sig Use a mirror for higher download bandwidth: https://www.gnu.o ... rg/order/ftp.html Here are the SHA256 and SHA3-256 checksums: File: parted-3.8.tar.xz SHA256 sum: a2b7811f47b0ddb1f7b1d0aa456f7c1270da70708ce231c2fe054c7199eafa63 SHA3-256 sum: dd794daa59755a9201d7cc493bcdf35903e231d48f93f7bdc79a780fcd8bf594 Verify the SHA256 checksum with either sha256sum, sha256, or 'shasum -a 256'. Verify the SHA3-256 checksum with 'cksum -a sha3 -l 256 --base64' 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 parted-3.8.tar.xz.sig parted-3.8.tar.xz The signature should match the fingerprint of the following key: pub ed25519/FF9868A2D488A5A9 2026-04-14 [SC] Key fingerprint = 872F 3A8A 0B84 905A FFBC 8766 FF98 68A2 D488 A5A9 uid [ultimate] Brian C. Lane <bcl@redhat.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 bcl@redhat.com gpg --recv-keys FF9868A2D488A5A9 wget -q -O- 'https://savannah. ... ed&download=1' | gpg --import - As a last resort to find the key, you can try the official GNU keyring: wget -q https://ftp.gnu.o ... u/gnu-keyring.gpg gpg --keyring gnu-keyring.gpg --verify parted-3.8.tar.xz.sig parted-3.8.tar.xz This release is based on the parted git repository, available as git clone https://https.git ... rg/git/parted.git with commit 5f600791b40a48c51aeb6db0432e67a29e8e7951 tagged as v3.8. For a summary of changes and contributors, see:

https://gitweb.gi ... a=shortlog;h=v3.8 or run this command from a git-cloned parted directory: git shortlog v3.7.14..v3.8 This release was bootstrapped with the following tools: Autoconf 2.72 Automake 1.18.1 Gettext 0.25.1 Gnulib 2026-09-23 0b416a8a26dbdaea4f413d4fb0c41ef8ea846e4d Gperf 3.2.1 NEWS Noteworthy changes in release 3.8 (2026-09-23) [stable] Promoting alpha release to stable release 3.8

- parallel @ Savannah: GNU Parallel 20260922 ('Parton') released [stable] (2026/09/22 19:36)
  GNU Parallel 20260922 ('Parton') has been released. It is available for download at: lbry://@GnuParallel:4 Quote of the month: GNU parallel is awesome. Use it more often in your scripts! -- Wade @WadeGrimshire@twitter New in this release: Bug fixes and man page updates. 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 ... rg/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.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/1 ... 81/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.

- automake @ Savannah: automake 1.19 released [stable] (2026/09/22 14:01)
  Automake 1.19 released. Announcement: https://lists.gnu ... -09/msg00001.html
- gzip @ Savannah: gzip-1.15 released [stable] (2026/09/20 08:05)
  This is to announce gzip-1.15, a stable release. Special thanks to Paul Eggert for his indefatigable support. There have been 119 commits by 5 people in the 75 weeks since 1.14. See the NEWS below for a brief summary. Thanks to everyone who has contributed! The following people contributed changes to this release: Bruno Haible (2) Collin Funk (2) Jim Meyering (24) Mark Adler (3) Paul Eggert (88) Jim [on behalf of the gzip maintainers] ========== Here is the GNU gzip home page: https://gnu.org/s/gzip/ Here are the compressed sources: https://ftp.gnu.org/gnu/gzip/gzip-1.15.tar.gz (1.2MB) https://ftp.gnu.org/gnu/gzip/gzip-1.15.tar.xz (792KB) https://ftp.gnu.org/gnu/gzip/gzip-1.15.zip (1.5MB) Here are the GPG detached signatures: https://ftp.gnu.org/gnu/gzip/gzip-1.15.tar.gz.sig https://ftp.gnu.org/gnu/gzip/gzip-1.15.tar.xz.sig https://ftp.gnu.org/gnu/gzip/gzip-1.15.zip.sig Use a mirror for higher download bandwidth: https://www.gnu.org/order/ftp.html Here are the SHA256 and SHA3-256 checksums: SHA256 (gzip-1.15.tar.gz) = VFiGz1f6iKZeln+/cFkD1/yyVnyCxzQkk+gujXsaIQs= SHA3-256 (gzip-1.15.tar.gz) = Pi3C560xPn/r7b8OPXzDZolBiOfpW/Bo9G+wUAdbgCw= SHA256 (gzip-1.15.tar.xz) = mqDMeA3sFWuCgoRIM7NCq3ywjCXSzZoYac3Q3zHe/0g= SHA3-256 (gzip-1.15.tar.xz) = gwwVDPignGjkxS/sw1QPN+PInnUE5sxAjkefL5C3/Mk= SHA256 (gzip-1.15.zip) = UvU0W04E/rexLC/qZ1/40GucNXMTtUh2yDu6cdICzhA= SHA3-256 (gzip-1.15.zip) = nhoi+h4j1rwxGJSA8pwkqehAoraNQGbcgvGxYOcxQkg= 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 gzip-1.15.tar.gz.sig gzip-1.15.tar.gz The signature should match the fingerprint of the following key: pub rsa4096/0x7FD9FCCB000BEEEE 2010-06-14 [SCEA] Key fingerprint = 155D 3FC5 00C8 3448 6D1E EA67 7FD9 FCCB 000B EEEE uid [ultimate] Jim Meyering <jim@meyering.net> uid [ultimate] Jim Meyering <meyering@fb.com> uid [ultimate] Jim Meyering <meyering@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 jim@meyering.net gpg --recv-keys 7FD9FCCB000BEEEE wget -q -O- 'https://savannah.gnu.org/project/release-gpgkeys.php?group=gzip&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 gzip-1.15.tar.gz.sig gzip-1.15.tar.gz This release is based on the gzip git repository, available as git clone https://https.git.savannah.gnu.org/git/gzip.git with commit e6263160768c73f6117f3c1de78291a60f2cb796 tagged as v1.15. For a summary of changes and contributors, see: https://gitweb.git.savannah.gnu.org/gitweb/?p=gzip.git;a=shortlog;h=v1.15 or run this command from a git-cloned gzip directory: git shortlog v1.14..v1.15 This release was bootstrapped with the following tools: Autoconf 2.73.1-b400b Automake 1.19 Gnulib 2026-09-04

331c8d065a8a753de71f084f068473ccd5e4c34a NEWS * Noteworthy changes in release 1.15 (2026-09-20) [stable] ** Bug fixes gzip no longer can mistakenly remove the wrong file if some other process simultaneously renames a gzip destination's ancestor. [bug present since the beginning] gzip -d no longer rejects PKZIP signatures, local header, and data descriptors. These can appear in well-formed streamed zip files. [bug present since the beginning] gzip diagnostics now quote file names containing unusual characters. [bug present since the beginning] A use of uninitialized memory on some malformed inputs has been fixed. [bug present since the beginning] A buffer overflow has been fixed when decompressing an .lzh file after decompressing a .Z file. [bug present since the beginning] When decompressing an .lzh file, the output is no longer corrupted when an internal bit buffer is not properly cleared. [bug present since the beginning] When decompressing an .lzh file after another .lzh file, the output is no longer corrupted by the previous file's decoding table. [bug present since the beginning] gzip --synchronous no longer fails to synchronize unreadable parent directories on platforms like GNU/Linux that have O_PATH, or to synchronize any parent directories on platforms like FreeBSD that have O_SEARCH but not O_PATH. [bug introduced in gzip-1.7] On old-fashioned or limited platforms lacking mktemp, gzexe, zdiff and znew no longer have a race when creating a temporary file. [bug present since the beginning] ** Changes in behavior gzip no longer insists on the "C" locale; instead, it follows the typical practice of using the locale specified by the environment. This change, which is needed for file name quoting, can affect the format of floating-point numbers output by gzip's -l and -v options. Diagnostics are still in English, though. gzip -l now reports "-Inf%" instead of "0.0%" for the infinite compression ratio of an empty file. znew's -P option is now ignored, with a warning. It was present only to improve performance, and its implementation had too many bugs to be worth supporting. ** Platforms no longer supported The following platforms (or earlier) are no longer supported because their old multibyte libraries do not work well enough: FreeBSD 4.11 (2005), HP-UX 11.00 (1997), Minix 3.1.8 (2010), MS-Windows 8.1 (2013) via mingw without UCRT.

- pspp @ Savannah: PSPP 2.1.2 has been released. (2026/09/19 21:23)
  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 2.1.1 to 2.1.2: Translation updates. Bug fixes. Please send PSPP bug reports to bug-gnu-pspp@gnu.org.
- www @ Savannah: What's Wrong with GitHub? (2026/09/16 21:57)
  A new article by Jacob Bachmeyer and Richard Stallman to answer the question, What's Wrong with GitHub? has been published at https://www.gnu.o ... -with-github.html Software developers, the GNU Project urges you to avoid hosting your repositories on GitHub. This is to avoid being on the receiving end of harmful practices, and avoid leading others to be victims too. This article focuses on a few of GitHub's worst problems, and the wrongs and harm they can do.
- unifont @ Savannah: Unifont 18.0.01 Released (2026/09/16 16:36)
  16 September 2026 Unifont 18.0.01 is now available. This release is aligned with Unicode 18.0.0, adding almost 400 new glyphs. Download this release from GNU server mirrors at: https://ftpmirror ... /unifont-18.0.01/ or if that fails, https://ftp.gnu.o ... /unifont-18.0.01/ or, as a last resort, ftp://ftp.gnu.org ... /unifont-18.0.01/ These files are also available on the unifoundry.com website: https://unifoundr ... /unifont-18.0.01/ Font files are in the subdirectory https://unifoundr ... 0.01/font-builds/ A more detailed description of font changes is available at https://unifoundr ... nifont/index.html and of utility program changes at https://unifoundr ... nt-utilities.html Information about Hangul modifications is at https://unifoundr ... hangul/index.html and http://unifoundry ... l-generation.html Enjoy! Paul Hardy, GNU Unifont

Maintainer

- GNU Health: GNUHealthCon 2026 – XI Free Software and Social Medicine Conference (2026/09/15 12:20)
  Dear community The XI edition of GNUHealthCon will take place in Gran Canaria, Spain, this December 18th, and you are invited! Celebrating 20 years of Freedom and Equity in Healthcare This edition is special for us because we will be celebrating the 20th anniversary of the first GNU Solidario mission that took place in Santiago del Estero, Argentina in October 2006. That remote rural school, the teachers, the children and their families generated a profound inspirational impact on me, so deep that since then I have dedicated most of my life to the field Social Medicine. GNU Health is both a result of that experience, and the main channel to deliver freedom and dignity around the globe. In this edition, besides the technical and social talks, we will have the space to commemorate these 20 years. During these two decades we have gone through many things. We have many happy stories to share, but we also have sad ones, that made us learn and be more resilient. Stories of people from countries around the world that have conformed this wonderful community around GNU Solidario. Stories and experiences that have made GNU Health the leading Free/Libre Health and Hospital Information System. I want to personally invite all of you who has been part of this beautiful journey: The GNU community; hospitals and health institutions around the world that use GNU Health; sister projects (Tryton, Orthanc,..); national and regional governments that have adopted GNU Health; sponsors; the open science and academic communities; developers and core team members... you are all part of the success of the project and we have to celebrate this edition together. Pre-conference party GHCON2026 will be on Friday, December 18th in Gran Canaria. The night before we will have the pre-conference party. Most probably, the event will be in a hotel in the mountains, and the idea is to arrive to the hotel on Thursday 17th. We will update in our official Mastodon account. (look for #GHCon2026 hashtag) Please make sure you register (https://www.gnuhealth.org/ghcon/2026/) so we can prepare de logistics. Looking forward to meeting personally all of you, and have a wonderful time in Gran Canaria! Love and happy hackingLuis
- FSF News: GNU Boot joins FSF fiscal sponsorship program (2026/09/14 21:55)
  BOSTON, Massachusetts, USA (Monday, September 14, 2026), â€” The Free Software Foundation (FSF) announced today that GNU Boot is its latest fiscally sponsored project. GNU Boot is a libre, ethical replacement for the nonfree BIOS or UEFI, which is software found in virtually all personal computers in the world today.
- coreutils @ Savannah: coreutils-9.12 released [stable] (2026/09/14 15:55)
  This is to announce coreutils-9.12, a stable release. Notable changes include: env supports a new --env0-from=FILE option to support full persistence and restoration of the environment. This also supports e.g. filtering like: env -i --env0-from=<(env -0 | sed -z ...) Commands now have safer terminal output, avoiding confusing output or corrupted terminal state. Commands that traverse directories no longer fail merely if files are being removed in parallel. ptx has improved robustness, avoiding potential infinite loops. tee fixes robustness issues introduced in the previous release, where it could go into an infinite loop or incorrectly error in the presence of short writes. stty is more accepting of variations in requested speed. sort(1) will now better use available memory and parallel operation when reading from pipes. There have also been many bug fixes and other changes as summarized in the NEWS below. There have been 288 commits by 16 people in the 21 weeks since 9.11. Thanks to everyone who has contributed! Arun Bhattacharya (1) Leonid Evdokimov (1) Ayesha Shafique (1) Mateusz Nosek (1) Bruno Haible (5) Max Downey Twiss (1) Collin Funk (98) Paul Eggert (12) Guanqiang Han (1) Pádraig Brady (118) H. Peter Anvin (1) Sylvestre Ledru (33) Ismail Ramzi (1) aizu-m (2) Iván Ezequiel Rodriguez (2) oech3 (11) 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-9.12.tar.gz (16MB) https://ftp.gnu.org/gnu/coreutils/coreutils-9.12.tar.xz (6.4MB) Here are the GPG detached signatures: https://ftp.gnu.org/gnu/coreutils/coreutils-9.12.tar.gz.sig https://ftp.gnu.org/gnu/coreutils/coreutils-9.12.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-9.12.tar.gz) = FMv1pN4Me3+jufp/raTFiy3v4zM2qo/YPXYixcTr3BM= SHA3-256 (coreutils-9.12.tar.gz) = 14aJyepDW60jsaL4cq5T/UFMwUJEkXDyUqZEQlsnLFM= SHA256 (coreutils-9.12.tar.xz) = pIAZhVlzPps9qZnpBUOsb4iKLKpUTY1mTFofF+Uo4hA= SHA3-256 (coreutils-9.12.tar.xz) = cKcyD1sxJr1tdWgWbh0+OKwOEIpaf2ACThlaHZooZDA= 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-9.12.tar.gz.sig coreutils-9.12.tar.gz 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-9.12.tar.gz.sig coreutils-9.12.tar.gz This release is based on the coreutils git repository, available as git clone https://https.git.savannah.gnu.org/git/coreutils.git with commit c0f8514d989184921d9b12a4d103a7b23abc5af8 tagged as v9.12. For a summary of changes and contributors, see: https://gitweb.git.savannah.gnu.org/gitweb/?p=coreutils.git;a=shortlog;h=v9.12 or run this command from a git-cloned coreutils directory: git shortlog v9.11..v9.12 This release was bootstrapped with the following tools: Autoconf 2.73.16-0a513 Automake 1.18.1 Gnulib 2026-09-07 106e9b2384d08a1696fcbd40cbab52237943f208 Bison 3.8.2 NEWS * Noteworthy changes in release 9.12 (2026-09-14) [stable] ** Bug fixes 'chcon', 'chgrp', 'chmod', 'chown', 'du', 'ls' which traverse hierarchies with -R, no longer fail merely because files may be being removed in parallel. [This bug was present in "the beginning".] 'comm - -' no longer closes standard input twice. Previously it would mistakenly exit with a nonzero status. [This bug was present in "the beginning".] 'cp', 'install', and 'mv' now fall back to a standard copy if a --reflink=auto clone fails due to EDQUOT, ENOMEM, or ENOSPC. E.g., with XFS, a clone can exhaust metadata space in an allocation group, while a standard copy works. [bug introduced in coreutils-9.2] 'cut -d' with multiple multi-byte delimiter options specified will correctly match the last delimiter specified. [bug introduced with multi-byte support in coreutils-9.11] 'date -d '1-2-3' "+%_D"' no longer propagates flags like _ and - to the year component of the %D date specifier, keeping consistent component formatting. [bug introduced in coreutils-8.31] 'du --max-depth=N' now exits with a nonzero exit status and an error message if N is negative. Previously it behaved as if N were zero. [bug introduced in coreutils-9.4] 'factor' avoids a buffer over-read (CWE-126) for certain values. [bug introduced in coreutils-9.8] 'head' and 'tail' now quote names in file headers when needed. [This bug was present in "the beginning".] 'ls --color' no longer reads freed memory when LS_COLORS sets "ln=target" and later becomes unparsable, e.g., LS_COLORS='ln=target:x'. [This bug was present in "the beginning".] 'mv' now warns when copying extended attributes fails with ENOTSUP, e.g., when moving files to a file system that does not support them. [bug introduced in coreutils-7.3] 'numfmt', 'printf', and 'seq' on Solaris, no longer output an extraneous e+00 when using a large precision like

"%.5119f". [This bug was present in "the beginning".] 'pinky -l' no longer no longer prints output in the incorrect order when standard output is fully buffered, e.g., when redirected to a file. [bug introduced in coreutils-9.10] 'pr' now exits gracefully upon exceeding internal accounting limits, like when processing large tab stops. [This bug was present in "the beginning".] 'ptx -G' no longer loops forever when the output width is smaller than twice the gap size, as with 'ptx -G -w4', or when a long reference leaves that little room, as with 'ptx -G -r'. [This bug was present in "the beginning".] 'ptx -W' no longer loops forever with a word regular expression that can match the empty string, like: echo ab | ptx -W 'a*'. [This bug was present in "the beginning".] 'shred' no longer blocks when opening a FIFO that has no readers. [This bug was present in "the beginning".] 'stty' no longer fails when the system uses speed encodings that are variations of the requested speed. [bug introduced in coreutils-9.8] 'tee' no longer loops infinitely after writing all output if a write call sets errno to EAGAIN. [bug introduced in coreutils-9.11] 'tee' no longer treats short writes as errors. [bug introduced in coreutils-9.11] 'test' no longer treats '-a' and '-o' as operators when given as strings to a binary operator. E.g., 'test -a -a -a' exits successfully instead of exiting with an error. [This bug was present in "the beginning".] 'truncate --reference=R' no longer hangs when R is a FIFO with no readers. [bug introduced in coreutils-8.17] 'unexpand -t' no longer overflows a heap buffer, for tab values > SIZE_MAX/16, or with multi-byte blank characters longer than the tab value. [bugs introduced in coreutils-9.11] 'uniq -w' no longer overruns the read buffer in multibyte locales. [bug introduced in coreutils-9.5] 'wc' no longer reads past the end of a lookup table in legacy multibyte locales like SHIFT-JIS where a single byte can decode to a wide character. [bug introduced in coreutils-9.5] Messages from Gnulib are no longer mistranslated in non-English locales. [bug introduced in coreutils-9.6] ** New Features 'env' now supports --env0-from=FILE to read NUL-delimited environment entries from a file. With -i, entries are preserved exactly, allowing full round-tripping of environments containing duplicate or nonstandard entries. 'stat' and 'tail' now know about the "failfs" and "nullfs" file system types. stat -f -c%T now reports the file system type, and tail -f uses inotify for these file systems. uname adds the -A,--all-labeled option to label all output, one item per line. ** Changes in behavior 'env' and 'printenv' now quote printed environment variables honoring the QUOTING_STYLE environment variable, defaulting to shell-escape style. This avoids printing arbitrary data to the terminal and allows the output to be sourced by a POSIX shell. 'ls' -w,--width no longer includes '\n' in the width of a line. I.e., the width or $COLUMNS is interpreted to be an _inclusive_ maximum. 'stat' now uses shell quoting when required, to more robustly escape file names. Previously it only quoted file names with the %N format. The default quoting honors the QUOTING_STYLE env variable (like %N). Also %Qn is a newly supported format combination to quote file names, leaving the existing %n format for when quoting is not desired. ** Improvements When built with the configure option '--with-wtmpdb', invocations of 'who /var/log/wtmp' and 'users /var/log/wtmp' use the wtmpdb database instead of the file /var/log/wtmp. This makes them Y2038-safe. 'cut -w' operates more efficiently when extracting the start of a line in multi-byte locales, giving 4x more throughput with typical input. 'df', 'du', 'ls', 'od', 'pr', and 'sort' now escape invalid arguments in error messages for options expecting an integer. 'env -0, and 'printenv -0' now explicitly set binary mode on output so that no CRLF translation is done e.g., on windows. 'install -C' will now avoid updating file metadata when the destination already has the appropriate ownership and permissions. 'basename', 'dirname', 'du', 'readlink', and 'realpath' now quote output in shell-escape style when standard output is a terminal. The QUOTING_STYLE environment variable can be used to adjust or disable the quoting. 'ls -m' now quotes files names containing commas when appropriate, so users can better distinguish separating commas. 'ls' now replaces newlines in file names if ambiguous with separators. Previously newlines were protected only when outputting to a terminal. 'sort' will now better use available memory and parallel operation when reading from unknown sized inputs like pipes. 'uniq -c' now operates up to 2.5x times faster on systems with unlocked stdio functions. ** Build-related 'logname' now builds, where

getlogin() is replaced (e.g. with musl), and systemd libs are being used, by linking the required libraries. The multi-call binary built with configure --enable-single-binary is reduced in size by around 10KB through the more efficient reuse of the 'test' code by '[', and the 'true' code by 'false'. configure no longer accepts the --with-linux-crypto option, which allowed cksum, md5sum, and sha*sum to use the Linux AF_ALG API. This API will be deprecated in Linux 7.2 and is less performant than OpenSSL. The configure option '--enable-systemd' is renamed to '--with-systemd'. The option '--enable-systemd' was a misnomer and is now deprecated. The 'sort' binary now uses the UAPI Group's .note.dlopen ELF note to indicate its dependency on libcrypto.

- GNU Guix: Demystifying complex configurations (2026/09/07 08:30)
  Guix system and Guix home introduce the concept of services. These provide users with a way to control background processes, commonly refereed as daemons, as well as ways of controlling the setup of files. For example, openssh-service-type is a service which controls a SSH daemon. In contrast, etc-service-type is a service that populates the contents of the /etc directory.One peculiarity of Guix services is that it's customary to provide Scheme bindings for the different fields. By that I mean that the different fields of the configuration of most services will be a type. The benefit of this is that users get a uniform configuration language for their services, at the cost of additional complexity when developing the service. Which is why, for a long time, Guix users seem to struggle with defining complex configurations. There are a number of reasons for this, we will try to close the gap today by going through defining a configuration for Goimapnotify.This blog post assumes that the reader is somewhat familiar with Guix and knows how to setup a development environment for it. If that's not the case, read The Perfect Setup and Using Guix interactively.A Goimapnotify configuration is written in YAML. Therefore, we will need to serialize the different Guile Scheme fields to this format. Let's take the example configuration that the author gives in the project's README.md:configurations: - host: example.com port: 143 tls: true tlsOptions: rejectUnauthorized: false starttls: true idleLogoutTimeout: 15 username: USERNAME alias: ExampleCOM password: PASSWORD xoAuth2: false boxes: - mailbox: INBOX onNewMail: 'mbsync examplecom:INBOX' onChangedMail: 'mbsync examplenet:INBOX' onChangedMailPost: SKIP onNewMailPost: SKIP - hostCMD: COMMAND_TO_RETRIEVE_HOST port: 993 tls: true tlsOptions: rejectUnauthorized: true starttls: true username: '' usernameCMD: '' password: '' passwordCMD: '' xoAuth2: false onNewMail: '' onNewMailPost: '' onChangedMail: '' onChangedMailPost: '' onDeletedMail: '' onDeletedMailPost: '' boxes: - mailbox: INBOX onNewMail: 'mbsync examplenet:INBOX' onNewMailPost: SKIP onChangedMail: 'mbsync examplenet:INBOX' - mailbox: Junk onNewMail: 'mbsync examplenet:Junk' onNewMailPost: SKIPJust by looking at the hierarchy we can already envision how to organize our scheme records. We will need the following:home-goimapnotify-configuration-fieldsgoimapnotify-configurationgoimapnotify-box-configurationgoimapnotify-tls-options-configurationThe final configuration that the service for Goimapnotify will rely on, will be home-goimapnotify-configuration-fields. It will contain a configurations field where each item will be a goimapnotify-configuration; each box in those configurations will be a goimapnotify-box-configuration. Additionally, each of those configurations will have an optional goimapnotify-tls-options-configuration.The (gnu services configuration) module provides us with the API that we need to define these configurations. The most important helpers for defining configurations are:define-configuration: For configurations that need to serialize fields into a different format, generally configuration files.define-configuration/no-serialization: For configurations that do not need to emit any files. Generally all the fields are consumed by the Shepherd services that rely on the configuration but no translation from Scheme to a different format is needed.In our case, we need to translate the different configuration fields to YAML, so we will need to use define-configuration.The following sections are written so you can follow along, you are encouraged to drop into a REPL (short for read-eval-print loop) and import the required module:,use (gnu services configuration)goimapnotify-tls-options-configurationTo make it easy for ourselves, we will

start form the in-out, after all, you wouldn't want to build a house from the rooftop, would you?(define-configuration goimapnotify-tls-options-configuration (reject-unauthorized? (boolean #f) "Whether to reject unauthorized TLS certificates.") (starttls? (boolean #f) "Whether to use STARTTLS.") (prefix goimapnotify-))evaluating the above snippet will throw an unbound variable exception, bear with me.Refer to the manual for an in-depth explanation of the syntax of define-configuration.The first argument is the name of the configuration object, goimapnotify-tls-options-configuration in this case. After it, we define the different typed fields. We are defining boolean fields that are, by default, set to false. The biggest source of confusion when defining configurations comes because define-configuration is a macro that introduces identifiers that do not appear in the source code—it's an unhygienic macro. That means that the macro will expand to code which defines symbols that are not visible when reading the source file.One can inspect the expansion of the macro by using the ,expand REPL command. It will be quite verbose, so don't try to read all of it, instead search through it; you will find some revealing things, such as:scheme@(gnu home services mail)> ,expand (define-configuration goimapnotify-tls-options-configuration (reject-unauthorized? (boolean #f) "Whether to reject unauthorized TLS certificates.") (starttls? (boolean #f) "Whether to use STARTTLS.") (prefix goimapnotify-)) $20 = (begin ... (define goimapnotify-tls-options-configuration? ...) (define goimapnotify-tls-options-configuration-reject-unauthorized? ...) (define goimapnotify-tls-options-configuration-starttls? ...) ... (define <goimapnotify-tls-options-configuration> ...) ... (define goimapnotify-tls-options-configuration ...) (define goimapnotify-tls-options-configuration-fields ((@@ (gnu services configuration) list) (let* ((name (let ((x 'reject-unauthorized?)) x)) ... (serializer (let ((x goimapnotify-serialize-boolean)) x)) ...) ...) (let* ((name (let ((x 'starttls?)) x)) ... (serializer (let ((x goimapnotify-serialize-boolean)) x)) ...) ...))))In the previous snippet, ... represents omitted code. As you can see from the macro expansion, define-configuration introduces quite a few identifiers. You may recognize the shared prefix in those identifiers, that's right, it's the prefix specified by that last field that we didn't explain from the configuration definition, (prefix goimapnotify-).By inspecting the macro expansion, it's easy to understand what happens under the hood. When a prefix is specified, we instruct the macro to append that prefix to all the generated identifiers. This is useful to avoid naming collisions; for example, when defining the configuration in a module with other configuration records that serialize to different formats. After all, it's not the same to serialize to YAML than to INI, or any other format a tool may require.Remember that unbound variable problem I mentioned before? If you paid close attention to the macro expansion, you may have noticed those references to goimapnotify-serialize-boolean, those are our unbound variables. The macro expects us to define these serialization procedures, let's do that.Defining the missing serializersSince the configuration machinery knows nothing about the output format, we must define how this translation happens. In our case, we are translating to YAML. By looking at the example configuration provided by the Goimapnotify developers, we can see that the field names are written in camel case, and their respective values are separated from the names through the : character.HelpersLet's start by making a function that takes a symbol and transforms it into a string that's a camelized version of that symbol. We will use object->camel-case-string from the (gnu home services utils) module:,use (gnu home services utils)(define (camelize-field-name field-name) (let ((str (object->camel-case-string field-name))) (if (string-suffix? "?" str) (string-drop-right str 1) str))) ;; Usage: (camelize-field-name 'reject-unauthorized?) => "rejectUnauthorized"In YAML, there is no convention of suffixing booleans with a ?, so our camelizer drops it when found.camelize-field-name gives us a field name, but we want to serialize the value too. Let's define a field serialization procedure to help us:(define (goimapnotify-serialize-field field-name val) "The mapping is used to serialize certain FIELD-NAMES specially." (let* ((field-name-mapping '((host-command . hostCmd) (user-name . username) (user-name-command . usernameCmd) (password-command . passwordCmd))) (field-name* (or (assq-ref field-name-mapping field-name) field-name))) (format #f "~a: ~s~%" (camelize-field-name field-name*) val))) ;; Usage: (goimapnotify-serialize-field 'reject-unauthorized? 'true) => "rejectUnauthorized: true\n"Notice how we

introduced field-name-mapping to tailor the field name passed to camelize-field-name to emit the specific naming that Goimapnotify expects. In Guix, we have a specific naming convention, so we want fields like user-name to map to username (instead of userName) and fields like password-command to map to passwordCmd (instead of passwordCommand).That's enough to serialize most values to YAML, but there's an extra Guix-specific feature we should make use of to make the service more convenient to users: G-Expressions.Are you familiar with them? If not, I encourage you to read this wonderful trilogy of blog posts: Dissecting Guix. It took me some time to wrap my head around these concepts, but once you do, I think you will like them too. In any case, that serializer is using those expressions because we want users of our configuration to be able to intermingle packages and other file-like objects in their fields. Since this is a blog post about writing configurations, I won't dive deep into G-Expressions, but let's try to get you a sense for them.You will need to import (guix gexp) for the following snippet to work.(define (goimapnotify-serialize-field field-name val) "The mapping is used to serialize certain FIELD-NAMES specially." (let* ((field-name-mapping '((host-command . hostCmd) (user-name . username) (user-name-command . usernameCmd) (password-command . passwordCmd))) (field-name* (or (assq-ref field-name-mapping field-name) field-name))) #~(format #f "~a: ~s~%" #$(camelize-field-name field-name*) #$val)))That's not so bad, is it? What was that? 6 more characters? Surely you wouldn't be scared of that, but just in case, let me give you some extra reassurance on what's happening here. This G-Expression thing, also known as gexp, is just the way you let Guix know that this code is for later. "When is later?", you may wonder. Simplifying it, that "later" is when Guix knows where file-like objects will be located in your disk; that long path on the store you may have seen before.For example, a package is a file-like object, so when you build the package Guix will tell you where it was stored:$ guix build cowsay /gnu/store/gz170ppi2hxssp4h3yw5jh4gdw6x9ws0-cowsay-3.8.4Preceding an expression with #~ (or gexp), one effectively tells Guix "don't evaluate this code until you know what this expression should expand to". And that other syntax, #$ (or ungexp), is telling Guix to replace the file-like object for its lowered representation, usually a path in the store. For a package like cowsay, that would be that /gnu/store/gz170ppi2hxssp4h3yw5jh4gdw6x9ws0-cowsay-3.8.4 paths we saw before.With that said, if you are like me, you won't be comfortable if you cannot debug this expressions, so let me give you some supper powers.In your REPL, import the (guix) module. This will augment the REPL with some additional commands that will simplify our life:scheme@(guile-user)> ,use (guix) scheme@(guile-user)> ,help guix Guix Commands [abbrev]: ,run-in-store EXP - Run EXP through the store monad. ,verbosity LEVEL - Change build verbosity to LEVEL. ,lower OBJECT - Lower OBJECT into a derivation or store file and return it. ,build OBJECT [BUILD-MODE] - Lower OBJECT and build it, returning its output file name(s). ,build-options OPTIONS - Set build options to OPTIONS. Print previous value (to allow easy restore). ,build-graft GRAFT? - Set whether grafts should be performed. ,enter-store-monad - Enter a REPL for values in the store monad. ,phases - Return the build phases of the package defined by FORM. ,configure-flags - Return the configure flags of the package defined by FORM. ,make-flags - Return the make flags of the package defined by FORM.Let's see what that gexp is up to:scheme@(gnu home services mail)> (goimapnotify-serialize-field 'reject-unauthorized? ''true) $6 = #<gexp (format #f "~a: ~s\n" #<gexp-input "rejectUnauthorized":out> #<gexp-input (quote true):out>) gnu/home/services/mail.scm:252:2 7f4c7d9deb40>There is a little helper to approximate a gexp to it's output:scheme@(gnu home services mail)> (gexp->approximate-sexp $6) $7 = (format #f "~a: ~s\n" "rejectUnauthorized" (quote true)) scheme@(gnu home services mail)> (primitive-eval $7) $8 = "rejectUnauthorized: true\n"That gets us an idea of what will be emitted to disk, but for more complex procedures, this is not going to cut it, specially if there are multiple gexps combined. So let's build that gexp:;; 'gexp->file' comes from the '(guix gexp)' module. You can ask the REPL more ;; information about a symbol through ',a SYMBOL'. scheme@(gnu home services mail)> ,a gexp->file (guix gexp): gexp->file #<procedure gexp->file (name exp #:key guile set-load-path? module-path splice? system target)> scheme@(gnu home services mail)> (gexp->file "test.scm" $6) $8 =

#<procedure 7f3f7d456ea0 at guix/gexp.scm:2098:2 (state)>In order for Guix to build a value, it needs to be something Guix can lower—an object that can be "compiled" down to a file in the store (these are referred to as file-like objects, because they can be inserted in any piece of code that expects a file name). A gexp cannot be built by itself, because there is nowhere to output it to. The above snippet creates a file-like object that will be emitted to a file named test.scm in the store. That file will contain our expression. As you see, we've got a procedure. The REPL printer tells us that this procedure needs some state; this is just a way of indicating you that this is a monadic procedure that can only be run in the context of a store connection. Read The Store Monad for more information.I will build it manually just once for demonstration purposes, don't blink:scheme@(gnu home services mail)> ,use (guix store) scheme@(gnu home services mail)> (run-with-store (open-connection) $8) $9 = #<derivation /gnu/store/gq71r3sgilfvsicfnifln8n0wfksam13-test.scm.drv => /gnu/store/c3hxw7pgnxchw1zldnc9nrb8iwwrqf78-test.scm 7f3f7e2de000>Where we are saying "run the procedure we got in the context of a store connection", we opened that connection through open-connection. Let's not do that again... Fortunately we have those useful REPL commands I just told you about, so we can archive the same result by doing this:scheme@(gnu home services mail)> ,run-in-store $8 $10 = #<derivation /gnu/store/gq71r3sgilfvsicfnifln8n0wfksam13-test.scm.drv => /gnu/store/c3hxw7pgnxchw1zldnc9nrb8iwwrqf78-test.scm 7fcd92f7f0f0>That's better, isn't it?Now, this is a derivation. That's something Guix can build:scheme@(gnu home services mail)> ,build $10 building /gnu/store/gq71r3sgilfvsicfnifln8n0wfksam13-test.scm.drv... $11 = "/gnu/store/c3hxw7pgnxchw1zldnc9nrb8iwwrqf78-test.scm"What's that? You don't want this magic command? Okay, the same thing can be done with:scheme@(gnu home services mail)> (run-with-store (open-connection) (built-derivations (list $10))) $12 = #t scheme@(gnu home services mail)> (derivation->output-path $10) $13 = "/gnu/store/c3hxw7pgnxchw1zldnc9nrb8iwwrqf78-test.scm"At last, a file! What's in there?$ cat /gnu/store/c3hxw7pgnxchw1zldnc9nrb8iwwrqf78-test.scm (format #f "~a: ~a\n" "rejectUnauthorized" "true")Okay that seems about right. We can even run it:scheme@(gnu home services mail)> (call-with-input-file $13 (lambda (port) (primitive-eval (read port)))) $14 = "rejectUnauthorized: true\n"You've seen the secret sauce, let's continue with what brought us here.(define (goimapnotify-serialize-boolean field-name val) (goimapnotify-serialize-field field-name (if val ''true ''false)))With this our configuration won't complain about the missing serializer. Let's continue.One last thing, notice how, after moving to the gexp version of goimapnotify-serialize-field, we started double quoting the symbols true and false, this is because the #$ syntax will replace the value in place, and the value of 'true is true, without the quote. If we didn't double quote, the staged code after expansion would look like this:(format #f "~a: ~s\n" "rejectUnauthorized" true)When what we really want is to expand to this:(format #f "~a: ~s\n" "rejectUnauthorized" 'true)The reason for this is that we want users to be able to write staged code on the different fields. If one of the values of a field where to be something like this:(goimapnotify-serialize-field 'favorite-game #~(string-append #$cowsay "/bin/cowsay"))The expansion would be this:(format #f "~a: ~s\n" "favoriteGame" (string-append "/gnu/store/gz170ppi2hxssp4h3yw5jh4gdw6x9ws0-cowsay-3.8.4" "/bin/cowsay"))The string-append procedure is evaluated right before the format call, not when the gexp is getting lowered.SerializingThe (gnu services configuration) module we imported earlier provides us with serialize-configuration, it takes two arguments, a configuration object and the fields that compose that configuration.The define-configuration macro defined the goimapnotify-tls-options-configuration constructor for us, we can create a configuration with the default values like this:scheme@(gnu home services mail)> (goimapnotify-tls-options-configuration) $10 = #<<goimapnotify-tls-options-configuration> reject-unauthorized?: #f starttls?: #f %location: #f>Then, we can serialize it like this:scheme@(gnu home services mail)> (serialize-configuration $10 goimapnotify-tls-options-configuration-fields) $11 = #<gexp gnu/services/configuration.scm:165:2 7ff1fd2cbf60>If you recall form the macro expansion we saw earlier, goimapnotify-tls-options-configuration-fields, was one of those identifiers that got generated.You should already know

how to build that gexp we've got:scheme@(gnu home services mail)> (gexp->file "test.yaml" $11) $12 = #<procedure 7efd8b10c2d0 at guix/gexp.scm:2098:2 (state)> scheme@(gnu home services mail)> ,use (guix) scheme@(gnu home services mail)> ,run-in-store $12 $13 = #<derivation /gnu/store/z59sf3nhh80668z8njljfxcymgi071pr-test.yaml.drv => /gnu/store/xam1210yl1vdxix0bgpfalf6drpv6xbx-test.yaml 7efd8a80fd70> scheme@(gnu home services mail)> ,build $13 $14 = "/gnu/store/xam1210yl1vdxix0bgpfalf6drpv6xbx-test.yaml" scheme@(gnu home services mail)> (call-with-input-file $14 (lambda (port) (display (primitive-eval (read port))))) rejectUnauthorized: false starttls: falseThat's some nice YAML syntax... We better speed up the pace or I will retire before finishing up this blog post.goimapnotify-box-configurationNext in line is goimapnotify-box-configuration, you know the drill. We start by declaring the configuration with define-configuration, specifying each field name, type and docstring:(define-configuration goimapnotify-box-configuration (mailbox string "The mailbox to monitor.") (on-new-mail maybe-string-or-gexp "The command to execute when new mail arrives.") (on-new-mail-post maybe-string-or-gexp "The command to execute after the new-mail command.") (on-changed-mail maybe-string-or-gexp "The command to execute when mail is changed.") (on-changed-mail-post maybe-string-or-gexp "The command to execute after the changed-mail command.") (on-deleted-mail maybe-string-or-gexp "The command to execute when mail is deleted.") (on-deleted-mail-post maybe-string-or-gexp "The command to execute after the deleted-mail command.") (prefix goimapnotify-))You may have noticed that we have two new types. strings, easy enough, and maybe-string-or-gexp, not so easy; right? Worry not, here comes the explanation.Let's start by defining the string type:(define-maybe string (prefix msmtp-configuration-))You may be very confused right now. What has msmtp-configuration to do with our goimapnotify-configuration example? Well, I want this blog post to get you ready for the real world, and in the wild, you will make configurations in modules that contain other configurations. The example we are looking up today is a narration of my adventures making the Goimapnotify service from the (gnu services mail) module, in that module, there is already a configuration for msmtp. With that said, if we tried to do this:(define-maybe string (prefix goimapnotify-))We would be surprised with a warning that looks something like this:$ make ... [94%] GUILEC gnu/home/services/mail.go gnu/home/services/mail.scm:67:0: warning: shadows previous definition of `maybe-string?' at gnu/home/services/mail.scm:63:0 gnu/home/services/mail.scm:278:0: warning: shadows previous definition of `goimapnotify-serialize-maybe-string' at gnu/home/services/mail.scm:67:0That's unexpected, isn't it? Nothing we've seen so far points to a maybe-string? procedure, let's look at what is going under the hood of that define-maybe macro:scheme@(gnu home services mail)> ,expand (define-maybe string (prefix msmtp-configuration-)) $15 = (begin (define (maybe-string? val) (or ((@@ (gnu services configuration) not) ((@@ (gnu services configuration) maybe-value-set?) val)) (string? val))) (define (msmtp-configuration-serialize-maybe-string field-name val) (if (string? val) (msmtp-configuration-serialize-string field-name val) "")))Do you see it? The macro is expanding to some code that introduces two new procedures into the module, one following the prefix, that would be msmtp-configuration-serialize-maybe-string, and one that just declares the predicate for the maybe type.Given that, the warning is now apparent. If we call again that macro with a string as the first argument, we will get the same predicate after the expansion, leading to the redefinition warning.So, for this particular case, instead of using the macro, we will define manually the two procedures required for our configuration:;; The module already had this helper defined. (define (string-or-gexp? obj) (or (string? obj) (gexp? obj))) ;; ... omitted lines ... (define (goimapnotify-serialize-string field-name val) (goimapnotify-serialize-field field-name val)) (define (goimapnotify-serialize-maybe-string field-name val) (if (maybe-value-set? val) (goimapnotify-serialize-string field-name val) "")) (define goimapnotify-serialize-string-or-gexp goimapnotify-serialize-string) (define (goimapnotify-serialize-maybe-string-or-gexp field-name val) (if (and (maybe-value-set? val) (string-or-gexp? val)) (goimapnotify-serialize-string-or-gexp field-name val) ""))The above snippet defines all the serializers we need for the new types. Notice that they are just some simple wrappers around the goimapnotify-serialize-field helper. That

procedure is doing all the heavy lifting here, and fortunately for us, it already handles gexps. Therefore, the fields that have a type that accepts a gexp are straightforward to declare.That would be it for the goimapnotify-box-configuration declaration. As showed in the previous section, the REPL is your friend. I had never written such a complex configuration before, but thanks to this "elegant weapon for a more civilized age", I could find my way by poking things around.Let's continue!goimapnotify-configurationYou know the drill, we start by defining the fields we need. Again, I'm doing this just by looking at the README example provided by the developers of Goimapnotify:(define-configuration goimapnotify-configuration (host string "The IMAP server hostname.") (host-command maybe-string-or-gexp "The command to retrieve the IMAP server hostname.") (port (integer 993) "The port that the IMAP server listens on.") (tls? (boolean #f) "Enable or disable TLS.") (tls-options maybe-goimapnotify-tls-options-configuration "TLS options for the IMAP connection." (serializer serialize-maybe-goimapnotify-tls-options-configuration)) (idle-logout-timeout maybe-integer "The idle logout timeout in minutes.") (user-name maybe-string "The user-name for authentication.") (user-name-command maybe-string-or-gexp "The command to retrieve the user-name.") (alias maybe-string "An alias for the account.") (password maybe-string "The password for authentication.") (password-command maybe-string-or-gexp "The command to retrieve the password.") (xo-auth2? (boolean #f) "Enable or disable XOAUTH2 authentication.") (wait maybe-integer "The delay in seconds before the mail syncing is triggered.") (boxes list-of-goimapnotify-boxes-configurations "The mailboxes to monitor." (serializer serialize-list-of-goimapnotify-boxes-configurations)) (prefix goimapnotify-))You are already familiar with most of those types, but there are some new things here:integer and maybe-integermaybe-goimapnotify-tls-options-configurationlist-of-goimapnotify-boxes-configurationsLet's start with the obvious ones first.integerThe module already had a maybe definition for the integer. If you recall from the macro expansion earlier, that means that there is already a symbol for the predicate maybe-integer?:(define-maybe integer (prefix msmtp-configuration-))We still need a serializer that follows our prefix:(define (goimapnotify-serialize-integer field-name val) (goimapnotify-serialize-field field-name val)) (define (goimapnotify-serialize-maybe-integer field-name val) (if (maybe-value-set? val) (goimapnotify-serialize-integer field-name val) ""))Simple enough. Moving on!maybe-goimapnotify-tls-options-configuration(define (serialize-goimapnotify-tls-options-configuration field-name val) (let ((serialization (serialize-configuration val goimapnotify-tls-options-configuration-fields))) #~(begin (use-modules (ice-9 format) (ice-9 string-fun)) (format #f "~a: ~a~%" '#$(camelize-field-name field-name) (string-replace-substring #$serialization "\n" "\n "))))) (define-maybe goimapnotify-tls-options-configuration)You are already familiar with the define-maybe macro. The serializer is also quite simple. Since we already defined a record that contains within all the required information to serialize it, we just need to make a simple wrapper around serialize-configuration.Before moving forward, notice how the serializers from this section doesn't contain that goimapnotify- prefix. This is an arbitrary decision, but since there is already goimapnotify in the symbol name, I think it's a bit redundant to add it. For the configuration definition to know which serializer to use, we have to be specific in the serializer argument of the fields, that's why in the configuration we had this declaration specifying a (serializer ...) for the field:(define-configuration goimapnotify-configuration ;; ... omitted lines ... (tls-options maybe-goimapnotify-tls-options-configuration "TLS options for the IMAP connection." (serializer serialize-maybe-goimapnotify-tls-options-configuration)) ;; ... omitted lines ... (prefix goimapnotify-))list-of-goimapnotify-boxes-configurationsFirst we need a prefix to know if we have a list of goimapnotify-box-configuration objects, that one is simple:(define (list-of-goimapnotify-boxes-configurations? lst) (and (not (null? lst)) (every goimapnotify-box-configuration? lst)))Now we need a serializer that knows how to handle a list of these objects:(define (serialize-list-of-goimapnotify-boxes-configurations field-name value) (let ((serializations (cons 'list (map (cut serialize-configuration <> goimapnotify-box-configuration-fields) value)))) #~(begin (use-modules (ice-9 format) (ice-9 string-fun)) (format #f "~a: ~{ - ~a~%~}" '#$(camelize-field-name field-name) (map (lambda (s) (string-replace-substring s "\n" "\n "))

#$serializations)))))Looks a bit daunting, but it's just staged code, remember our earlier explanation of gexps. This code is only constructing a string with the shape we need.As a reminder, we are serializing to YAML, that means that lists are prefixed by a - character. Following the Goimapnotify README, the indentation would be something like this:boxes: - mailbox: INBOX onNewMail: 'mbsync examplenet:INBOX' onNewMailPost: SKIP onChangedMail: 'mbsync examplenet:INBOX' - mailbox: Junk onNewMail: 'mbsync examplenet:Junk' onNewMailPost: SKIPThis is what that format call is doing. Refer to Formatted-Output for more information.We are only lacking a way to generate a complete configuration, remember that goimapnotify-configuration is an object for a single configuration. According to the README of Goimapnotify, the configuration file can take a list of configurations.home-goimapnotify-configurationThis is the last configuration we will need, it will be used directly by the service:(define-configuration home-goimapnotify-configuration (goimapnotify (file-like goimapnotify) "The @code{goimapnotify} package to use." empty-serializer) (configurations (list-of-goimapnotify-configurations) "A list of @code{goimapnotify-configuration} records which contain information about all your accounts configurations."))Simple enough, the first field is the Guix package that provides the goimapnotify program. This one is used by the service to start the process. Since it doesn't need to appear in any configuration file, we don't need to serialize it, hence the empty-serializer.On last serializer:(define (serialize-list-of-goimapnotify-configurations field-name value) (let ((serializations (cons 'list (map (cut serialize-configuration <> goimapnotify-configuration-fields) value)))) #~(begin (use-modules (ice-9 format) (ice-9 string-fun)) (format #f "~a: ~{ - ~a~%~}" '#$(camelize-field-name field-name) (map (lambda (s) (string-replace-substring s "\n" "\n ")) #$serializations)))))The rationale for this code is the same as the one explained for list-of-goimapnotify-boxes-configurations.Defining the Shepherd serviceThe service definition; at last!(define (home-goimapnotify-shepherd-service config) (let ((log-file #~(string-append %user-log-dir "/goimapnotify.log"))) (list (shepherd-service (provision '(goimapnotify)) (modules '((shepherd support))) ;for '%user-log-dir' (documentation "Run a goimapnotify process") (start #~(make-forkexec-constructor (list #$(file-append (home-goimapnotify-configuration-goimapnotify config) "/bin/goimapnotify") "-conf" #$(mixed-text-file "goimapnotify.yaml" (serialize-configuration config home-goimapnotify-configuration-fields))) #:log-file #$log-file)) (stop #~(make-kill-destructor))))))Refer to Shepherd Services for more information on how to write Shepherd services. I will only highlight how to handle the configuration we just wrote.We have already done all the hard work, the configuration declaration contains all the information needed to perform the serialization of the different fields to YAML, we just need to call serialize-configuration. For example:scheme@(gnu home services mail)> (define test-config (home-goimapnotify-configuration (configurations (list (goimapnotify-configuration (host "test.example.com") (boxes (list (goimapnotify-box-configuration (mailbox "Test")))))))) scheme@(gnu home services mail)> (serialize-configuration test-config home-goimapnotify-configuration-fields) $6 = #<gexp gnu/services/configuration.scm:165:2 7f3448e678a0>That serialization gives a gexp ready to be wrapped in a file-like so it can be lowered to the store. That's what mixed-text-file is doing:scheme@(gnu home services mail)> (mixed-text-file "goimapnotify.yaml" (serialize-configuration test-config home-goimapnotify-configuration-fields)) $7 = #<<computed-file> name: "goimapnotify.yaml" gexp: #<gexp guix/gexp.scm:2171:6 7f343915ef60> guile: #f options: (#:local-build? #t)> scheme@(gnu home services mail)>We've got ourselves a service, but there is one last thing before we go!Defining the service typeAs mentioned in the previous section, refer to Defining Services for what's going on here. The last thing we need do is declare the relation that this service has with respect to others:(define home-goimapnotify-service-type (service-type (name 'home-goimapnotify) (extensions (list (service-extension home-shepherd-service-type home-goimapnotify-shepherd-service))) (description "Configures the @code{goimapnotify} IMAP Mailbox notifier.")))We did it!Closing wordsStill here? That was long... But here we are, we defined our complex configuration.Congratulations on reading till the end, by this time you should already be an expert on defining Guix configurations.We

acknowledge that there are some improvements to do on the API for defining configurations. We would like to have a way to specify symbol mapping for field names in the declaration, so that serializers can be generalized better. It would also be nice if we could refactor some of our configuration definitions so the serializers that can be generalized are reused between declarations. After all, there are many configuration files in similar formats.For all of these improvements, we are counting on you! I wrote this blog post to empower you to participate in the development. If this is something that resonates with you, come join the fun!

- Simon Josefsson: Soft-launching the DiffOS project (2026/09/04 13:11)
  Today marks the day of soft-launching of my Debian derivative, which I've been using on several of my own machines for the past year or so. This is still work in progress, but I wanted to establish a launch date of the project so below is the DiffOS manifesto as motivation for continued work. DiffOS is For Freedom! DiffOS is the Debian Increment For Freedom Operating System. Aspire to the goals of GNU FSDG and become a recognized Free GNU/Linux distribution. Uses Debian GNU/Linux as upstream. Support for all architectures supported by Debian. Provide Containers, Cloud Images, LiveCD and installer ISOs. Provide standalone hosting of the package repository. Provide documentation and issue tracker. Keep changes to a minimal, in particular: Upstream-first policy to prefer that any changes are made in Debian, and only if that fails they are considered for DiffOS. Binary package re-use for as much as is possible. Don't modify any source-level Debian package unless REQUIRED by the FSDG (e.g., for freedom concerns) or REQUIRED by the Debian project (e.g., for branding reasons). Publish a list of packages that are added, removed or modified compared to Debian, with justification for each change. Publish Diffoscope-style outputs comparing our artifacts with comparable Debian artifact. Everything built from CI/CD pipelines, inspired by the Salsa CI pipeline but extended to cover the package repository and installation images as well, to allow modern GitSecDevOps of the entire supply-chain. Use inspiration from other Debian-derived FSDG distributions Trisquel GNU/Linux and PureOS, and broader with GNU Guix especially on how to approach existing freedom concerns in packages. Git Forge agnostic. While currently hosted on GitLab.com, scripts and configuration are (or will be) designed to allow setup on self-hosted GitLab instance, Codeberg.org or self-hosted Forgejo. Maintained by Humans – THE HUMAN MANIFESTO FOR THE AGE OF ARTIFICIAL INTELLIGENCE. Happy Hacking!
- parallel @ Savannah: GNU Parallel 20260822 ('Ceuta') released (2026/08/30 21:07)
  GNU Parallel 20260822 ('Ceuta') has been released. It is available for download at: lbry://@GnuParallel:4 Quote of the month: Parallel has an option for almost everything, it's almost too much. -- tester457@ycombinator New in this release: Replacement string: {%jq: jq-expression %} Use $^X when calling perl. So if different versions of perl are found in $PATH, we use the same version that started GNU Parallel. Bug fixes and man page updates. 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 ... rg/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.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/1 ... 81/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.

- freeipmi @ Savannah: FreeIPMI 1.6.19 Released (2026/08/28 16:21)
  o Fix minor groff warnings in manpages. o Fix portability of building manpages. o Fix minor bugs found by code analysis: - bmc-watchdog: Fix bug with --arp-response command line parsing. - ipmi/rmcpping: Fix bug with finding an IPv6 interface. - ipmidetect: fix bug in which hostname configs not used properly - ipmi-oem intelnm: fix parsing of hours/minutes error - ipmi-oem: Correct Dell CMC IPv6 autoconfiguration output. - ipmi-oem: Correct Dell iDRAC web server control output. - libipmidetect: Isolate partial results between fallback servers. - libipmidetect: Honor caller-provided hostnames over configured defaults. - common: Preserve stream state when finishing buffered output. - common: Fix buffer-output configuration parse issue. - common: Parse UTC offset configuration as an integer. - ipmiconsole: Monitor both console descriptors. - ipmiconsole: Check valid payload instance range correctly. - ipmi-sel: Honor post-clear after full tail output. - ipmi-fru: Report DIMM capacities in megabytes. - ipmi-chassis: Apply the Power-On Hours scale correctly. - libipmimonitoring: fix mem-leak on SEL iterator - libipmimonitoring: allow cipher suite 0 in configs - libipmimonitoring: report NO_SEL_RECORDS instead of NO_SENSOR_READINGS cut and paste errors in some functions. o Fix potential stack overflows found by code analysis in ipmi-oem dell get-system-info command (specifically idrac-info, cmc-info,

cmc-ipv6-info subcommands) and ipmi-oem fujitsu get-sel-entry-long-text. o Fix potential stack overflow in libfreeipmi also related to Fujitsu long SEL entries. https://ftp.gnu.o ... pmi-1.6.19.tar.gz

- FSF News: Job opportunity: Systems Integration Administrator at the Free Software Foundation (2026/08/26 18:55)
  BOSTON, Massachusetts, USA (Wednesday, August 26, 2026) â€" 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 Systems Integration Administrator. This position is ideally full-time and US-based, but exceptions can be made for a qualified candidate.
- automake @ Savannah: automake-1.18.92 released [beta] (2026/08/25 14:32)
  This is to announce automake-1.18.92, a beta release in preparation for automake-1.19. Announcement: https://lists.gnu ... -08/msg00000.html
- unrtf @ Savannah: unrtf 0.21.12 (2026/08/24 08:22)
  unrtf 0.21.12 is released, fixing a serious recently submitted security issue.
- GNU Taler news: Book chapter on Taler as sCBDC technology published (2026/08/20 22:00)
  The chapter is published as DOI 10.1007/978-3-032-22946-5_9 and the author's preprint is available here. The abstract reads as follows:
- GNUnet News: GNUnet 0.29.0 (2026/08/19 22:00)
  GNUnet 0.29.0 released We are pleased to announce the release of GNUnet 0.29.0. 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. The release addresses a couple of regressions and general instability of the transport subsystem. Major versions may break protocol compatibility with the 0.28.X versions. Please be aware that Git master is thus henceforth (and has been for a while) INCOMPATIBLE with the 0.28.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 0.29.0 release is still only suitable for early adopters with some reasonable pain tolerance . Download links gnunet-0.29.0.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 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.
- health @ Savannah: GNU Health strict No Generative Artificial Intelligence Policy (2026/08/19 15:30)
  Dear community We have included in the GNU Health Code of Conduct the strict NO Generative Artificial Intelligence policy. The current version reads: GNU Health Strict No Generative Artificial Intelligence Policy GNU Health is social project made by humans and for humans. We DO NOT accept any code, artwork, review, documentation or issues created by generative Artifical Intelligence (GenAI) / Large Language Models (LLMs).

The GNU Health no-AI policy is because we strongly believe that: * GenAI is bad for Mother Nature * GenAI is bad for human rights, especially for underserved and marginalized communities. * GenAI is bad for the Free Software and Free Culture communities. * GenAI is bad for you Last but not least, GNU Health manages critical health information both at personal and population level. There must be a reasoning behind every single line of code. We make all the effort to minimize bugs that can jeopardize the integrity and security of the system, and we can not risk the project by putting it in hands of stochastic parrots. Letâ€™s keep the art and science of computing a human virtue. You can read the entire, most current version of GNU Health Code of conduct here: https://docs.gnuh ... ndix/conduct.html

- poke @ Savannah: GNU poke 5.0 released (2026/08/17 08:51)
  I am happy to announce a new major release of GNU poke, version 5.0. GNU poke 5.0 release is now available at https://ftp.gnu.o ... e/poke-5.0.tar.gz The tarball is signed and you can get the PGP signature at https://ftp.gnu.o ... ke-5.0.tar.gz.sig GNU poke (http://www.j ... rch.net/poke) is an interactive, extensible editor for binary data. Not limited to editing basic entities such as bits and bytes, it provides a full-fledged procedural, interactive programming language designed to describe data structures and to operate on them. I'd like to thank everyone who contributed to this release through code, documentation, or testing. What is new in this release: User interface updates Now hyperlink server can bind to a user-specified port for listening to commands (-p, --hserver-port). Poke Language updates Floating-point arithmetic is now supported on uint<32>/uint<64> types. uint<32> will be interpreted as a single-precision floating-point number and uint<64> will be interpreted as a double-precision floating-point number as defined per the IEEE 754 standard. The following expressions are now supported: - Addition: a .+ b - Subtraction: a .- b - Multiplication: a .* b - Division: a ./ b - Ceil-devision: a ./^ b - Exponentiation: a .** b - Remainder: a .% b - Post-increment: a.++ - Pre-increment: .++a - Post-decrement: a.-- - Pre-decrement: .--a - Negation: .-a - Less-than: a .< b - Less-than-or-equal-to: a .<= b - Greater-than: a .> b - Greater-than-or-equal-to: a .>= b - Equal-to: a .== b - Not-equal-to: a .!= b Poke Runtime updates Thanks to the great work of David Faust, poke now supports reactive IO spaces! Extent of a PVM value mapped in a given IO space will be tracked and values will be re-mapped only if a write happens in their extent; which is a big performance win for read-intense programs. A bunch of undefined behavior (UB) releated to left-shifts has been fixed. Improved human-readable message of E_conv exception when verifying length/size of an array with dynamic bound(s) to help the user to understand the mistake. Poke compiler updates Now poke can properly handle writes to nested integral struct/unions fields. Previously write to nested fields of integral structs did not materialize in IO space. Standard Poke Library updates Closure's pretty printer now adds closure's name (identifier) to output. Two new functions to calculate square root of single and double precision floating point numbers: sqrtf and sqrtd. They accept uint<32> and uint<64> respectively as the IEEE 754 single and double precision floating-point numbers. libpoke updates Version of DSO is bumped to 2.0.0, and from this release onward, we try to not break the ABI, and bump the version components according to the libtool's recommendation (when needed). To be able to keep the ABI backward-compatibility promise, all public APIs are now accepting either pk_compiler or pk_val. This is the first step toward removing global state from libpoke to be able to have multiple instances of libpoke in a single process (and also to be able to accomplish thread-safety). We're not there yet, but we'll be there some day (hopefully soon)! IO subsystem updates IOS_F_TRUNCATE has been re-introduced (it was removed after release of poke 1.0 by the rationale that it's not that useful of a flag. Turns out it's quite useful to start from an empty file when assembling binary files from scratch using poke. Pickles updates Improved ustar pickle and add tests. Method get_last_mod_time has been fixed and the following methods has been added: get_{file,owner_user,group}_name. Improved time pickle to print date and time properly (zero-padded), and also add ptime_str function to get

date/time information as a string. Platform supports We tried to improve MinGW compilation situation by importing more Gnulib modules, but we still cannot have poke executable for MinGW platform. Help is very much appreciated in this area! Documentation updates Thanks to people who actually read the reference manual, this release includes a bunch of corrections to the documentation! Cheers to them! Happy poking! Mohammad-Reza Nabipoor

- wdiff @ Savannah: wdiff-1.2.3 released [stable] (2026/08/12 19:28)
  This is to announce wdiff-1.2.3, a stable release. There have been 29 commits by 2 people in the 637 weeks since 1.2.2. See the NEWS below for a brief summary. Thanks to everyone who has contributed! The following people contributed changes to this release: Bob Proulx (6) Simon Josefsson (23) Happy Hacking, Simon [on behalf of the wdiff maintainers] ========== Here is the GNU wdiff home page: https://www.gnu.org/software/wdiff/ Manual: https://www.gnu.org/software/wdiff/manual/ https://www.gnu.org/software/wdiff/manual/wdiff.html - HTML format https://www.gnu.org/software/wdiff/manual/wdiff.pdf - PDF format Here are the compressed sources and a GPG detached signature: https://ftp.gnu.org/gnu/wdiff/wdiff-1.2.3.tar.gz https://ftp.gnu.org/gnu/wdiff/wdiff-1.2.3.tar.gz.sig Here is minimal source-only "git archive" sources: https://ftp.gnu.org/gnu/wdiff/wdiff-v1.2.3-src.tar.gz https://ftp.gnu.org/gnu/wdiff/wdiff-v1.2.3-src.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: wdiff-1.2.3.tar.gz SHA256 sum: 29a4457eb0ed35c902e6732d71f25e1d6c7fe7fa0eda0fb6c371ed6779b49fd6 SHA3-256 sum: 2f558fe6b7a69524f29d266c4018c80a10135cef2e4a53ae3c2944c1fc12c95d File: wdiff-v1.2.3-src.tar.gz SHA256 sum: dc29d8f530471b9a8935413dca32e45848363f6e58888fc7307f5f052b8fb827 SHA3-256 sum: 4db1f41eda5e1525f3ced491af1fc1ad094cc21c209e789d12a1ea8f611be24e Verify the SHA256 checksum with either sha256sum, sha256, or 'shasum -a 256'. Verify the SHA3-256 checksum with 'cksum -a sha3 -l 256 --base64' 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 wdiff-1.2.3.tar.gz.sig The signature should match the fingerprint of the following key: pub ed25519 2019-03-20 [SC] B1D2 BD13 75BE CB78 4CF4 F8C4 D73C F638 C53C 06BE uid Simon Josefsson <simon@josefsson.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 simon@josefsson.org gpg --recv-keys 51722B08FE4745A2 wget -q -O- 'https://savannah.gnu.org/project/release-gpgkeys.php?group=wdiff&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 wdiff-1.2.3.tar.gz.sig This release is based on the wdiff git repository, available as git clone https://https.git.savannah.gnu.org/git/wdiff.git with commit 31fe62c4a3e2a76e5d68f571b1aa2f10815ca51f tagged as v1.2.3. For a summary of changes and contributors, see: https://gitweb.git.savannah.gnu.org/gitweb/?p=wdiff.git;a=shortlog;h=v1.2.3 or run this command from a git-cloned wdiff directory: git shortlog v1.2.2..v1.2.3 This release was bootstrapped with the following tools: Gnulib 31fe62c4a3e2a76e5d68f571b1aa2f10815ca51f Autoconf 2.72 Automake 1.17 Make 4.4.1 Makeinfo 7.3 Help2man 1.49.2 Tar 1.35 Gzip 1.13 NEWS * Noteworthy changes in release 1.2.3 (2026-06-30) [stable] ** Modernize build environment. Several minor fixes including build fixes for modern C23 compilers. Many generated or obsolete files have been removed from git.
- poke @ Savannah: GNU poke 4.90 pre-release is available (2026/08/10 22:33)

GNU poke (http://www.j ... rch.net/poke) is an interactive, extensible editor for binary data. Not limited to editing basic entities such as bits and bytes, it provides a full-fledged procedural, interactive programming language designed to describe data structures and to operate on them. GNU poke 4.90 pre-release is now available at https://alpha.gnu ... /poke-4.90.tar.gz The tarball is signed and you can get the PGP signature at https://alpha.gnu ... e-4.90.tar.gz.sig The planned date for releasing 5.0 is Sunday 15 August 2026, but this may change depending on the amount of problems found in this pre-release, and the subsequent needed additional pre-releases. Please report any problem found with the pre-release, comments or patches to poke-devel@gnu.org. In behalf of the poke developers, thank you! Happy testing! Mohammad-Reza Nabipoor

- FSF News: FSF publishes incident report on GNU Savannah vulnerabilities (2026/08/07 21:57)
- Greg Casamento: Open Source Is Hobbling Itself Over Generative AI (2026/08/04 10:52)

  The answer to bad AI-assisted contributions is not a purity test. It is better engineering discipline.Earlier this year, a discussion in the GNUstep community raised a proposal that will sound familiar across the Free Software world: prohibit AI-generated code in core projects and proudly advertise the result as “coded by humans” or “AI-free.” The argument was not frivolous. Generative AI raises real questions about copyright, attribution, security, energy use, labor, trust, and the flood of low-quality patches that maintainers are increasingly being asked to review.But a blanket refusal to use generative AI is the wrong response. It does not solve the hardest problems. It creates rules that are nearly impossible to define or enforce, confuses the method of production with the quality of the product, and risks turning Free Software into a movement that protects yesterday’s workflow instead of protecting software freedom.Open Source and Free Software are already operating with too few maintainers, too much technical debt, and too many important projects resting on the unpaid labor of a handful of people. We should be very careful about categorically rejecting tools that might help contributors understand old code, write tests, improve documentation, port software, find defects, or perform mechanical modernization. We should be even more careful when our proposed alternative offers the appearance of trust without the substance of it.The better principle is straightforward:Regulate the code, not the development process.“AI-generated” is not a workable boundaryWhat exactly counts as AI-generated code?Is it a complete function produced from a prompt? A line accepted from an AI-powered autocomplete system? A compiler-suggested correction? An automated refactoring? A test generated from an existing implementation? A translation of documentation? A patch written by a human after asking a model to explain an unfamiliar API? What if the developer uses AI to identify the problem but writes every line manually? What if an IDE quietly includes machine-learning features the contributor never explicitly invoked?The line between “human-written” and “AI-assisted” is already blurred, and it will become less distinct as generative features are embedded in editors, compilers, debuggers, search engines, and operating systems. A ban that cannot draw a stable boundary will be applied inconsistently. Honest contributors will disclose and be penalized; dishonest contributors will simply omit the disclosure. Others may be falsely accused because their code “looks generated.”An “AI-free” badge therefore risks promising something a project cannot reliably prove. Free Software should be especially suspicious of unverifiable labels.The risks are real—and they argue for reviewNone of this means generated code should be trusted.Research has found substantial security weaknesses in AI-produced code. One empirical study of Copilot snippets found security problems in roughly 30 percent of Python snippets and 24 percent of JavaScript snippets in its later dataset. Other research has demonstrated that code models can memorize portions of their training data, while studies of license compliance have found that models often provide inaccurate licensing information, particularly for copyleft code. Those are serious concerns, not anti-AI superstition. (Security weaknesses study; memorization study; license-compliance study)The productivity story is also more complicated than the advertising. GitHub reported that developers completed a controlled programming task considerably faster with Copilot, but a later randomized study of experienced Open Source

developers working in their own repositories found that the tools available in early 2025 made them 19 percent slower. METR’s 2026 follow-up found suggestive but still statistically uncertain evidence of improvement with newer tools. AI is neither magic nor uniformly useless; its value depends on the person, task, model, and workflow. (GitHub productivity study; METR 2025 study; METR 2026 update)But human authorship has never guaranteed secure, original, maintainable, or correctly licensed code. That is why healthy projects require tests, review, contributor certification, licensing rules, and maintainers who can reject bad work. The origin of a patch may affect how carefully we inspect it, but it cannot replace inspection.If a contributor submits code they do not understand, the contribution should be rejected. If the patch fails tests, violates project style, invents APIs, introduces vulnerabilities, obscures provenance, or imposes an unreasonable review burden, it should be rejected. That is true whether the patch was produced by Claude, Copilot, a Stack Overflow answer, a contractor, a junior programmer, or a senior maintainer having a bad afternoon.The repository contains code, not virtue.Review capacity is the scarce resourceMaintainers have a legitimate complaint: AI can make producing a patch far cheaper than reviewing one. A person can generate thousands of lines in minutes and then expect a volunteer to spend hours establishing whether any of it is correct. That asymmetry can become a denial-of-service attack on a project even when the submitter means well.The answer, however, is not necessarily to ban a tool. It is to place the cost and responsibility back on the contributor.A project can require that contributors:disclose material use of generative AI;identify the tool and describe how it was used;certify that they reviewed and understand every submitted change;explain the design and answer maintainer questions without outsourcing the conversation to a model;provide focused tests and evidence that the patch solves a real problem;comply with the project’s licensing and provenance requirements;keep changes small enough to review; andaccept that unexplained, low-signal, or mass-generated submissions may be closed without detailed triage.Disclosure is imperfect, but it establishes a community norm and makes an honest contributor accountable. Research into self-declaration practices has already found developers using everything from a simple disclosure to records of prompts, explanations, and quality checks. Projects can choose a level proportionate to their risk. (Study of AI-code self-declaration)This approach is stricter than either blind enthusiasm or symbolic prohibition. It does not say, “AI wrote it, so it must be acceptable.” It says, “You submitted it, so you are responsible for it.”Freedom is not a reenactment of an older toolchainFree Software is founded on the user’s freedom to run, study, modify, and share software. Those principles describe control over technology; they do not require that every developer use the same approved method to create it. The Open Source Initiative’s work on an Open Source AI Definition likewise frames the issue around the practical freedoms to use, study, modify, and share systems—not around preserving a pre-AI development ritual. (Open Source AI Definition 1.0)There are valid reasons for preferring Free or locally operated AI tools over proprietary cloud services. A project may reasonably prohibit contributors from uploading confidential material or unreleased security fixes to third-party systems. It may impose stricter provenance requirements in sensitive components. Individual maintainers may decline to review bulk-generated reports that have repeatedly produced noise. These are concrete policies tied to concrete harms.What does not follow is that a project becomes more free merely because no contributor used a generative tool.An “AI-free” identity may even distract from the qualities that users actually need: portability, stability, compatibility, security, good documentation, responsive maintenance, and code whose behavior can be understood and changed. A badge is not a substitute for those things.Blanket refusal has an opportunity costMature Free Software projects often contain decades of code and institutional knowledge. They need documentation, regression tests, API audits, build-system repairs, platform ports, translations, issue triage, and repetitive modernization. Generative AI will not perform those jobs reliably on its own. It can still help a knowledgeable contributor perform some of them.Rejecting that possibility at the policy level has consequences. It may discourage younger contributors whose development environment already includes these

tools. It may disadvantage people working in a second language or developers with disabilities who use AI as an accessibility aid. It may prevent experiments that would have failed harmlessly—or succeeded usefully—under ordinary review. Most dangerously, it can encourage a culture in which the declaration “human-written” is treated as evidence of quality.Free Software has survived previous waves of automation. High-level languages, garbage collection, IDEs, graphical interface builders, code generators, automated formatters, static analyzers, and online code search all changed what it meant to “write” software. Each tool altered the division of labor between programmer and machine. The relevant question was never whether every token originated in a human mind. The question was whether people retained the freedom, knowledge, and responsibility needed to control the resulting system.That remains the right question now.A policy that protects projects without freezing themA sensible policy can fit on one page:Disclosure: Contributors must disclose material AI assistance in the commit message or pull request.Responsibility: The named human contributor is the author of record and must understand, explain, test, and stand behind the entire submission.Quality: AI-assisted contributions receive the same requirements for correctness, security, maintainability, style, documentation, and test coverage as any other contribution.Provenance: Contributors must have a reasonable basis to believe the submission is license-compatible and must identify known sources or generated passages that may reproduce existing code.Data protection: Project secrets, embargoed vulnerabilities, private communications, and other restricted material may not be submitted to unauthorized external services.Reviewability: Maintainers may reject oversized, unexplained, repetitive, or low-signal submissions without performing free forensic work for the submitter.Local discretion: Components with unusual legal, safety, privacy, or reliability risks may adopt additional written restrictions.This policy does not resolve every ethical question surrounding generative AI. No contribution policy can. It does, however, address the matters a software project can actually evaluate and enforce.We should not surrender the future of software freedomThe Free Software community should remain one of the sharpest critics of concentrated corporate power, opaque models, exploitative data practices, environmental cost, and systems that deprive users of control. Criticism is part of our job. So is building an alternative.If we define ourselves by refusing to touch an important new class of technology, proprietary vendors will shape that technology without us. If instead we insist on transparency, modifiability, privacy, local control, licensing clarity, and human accountability, we can bring the values of Free Software into the AI era.We do not need to pretend that generative AI is trustworthy. We need processes that do not require us to trust it.Judge the patch. Demand disclosure. Require understanding. Enforce licensing. Protect reviewers. Reject garbage.But do not hobble Open Source and Free Software with a blanket ban that is difficult to define, impossible to verify, and disconnected from the quality of the code we ultimately ship.

- libc @ Savannah: The GNU C Library version 2.44 is now available (2026/07/27 14:36)
  The GNU C Library ========== The GNU C Library version 2.44 is now available. The GNU C Library is used as the C library in the GNU system and in GNU/Linux systems, as well as many other systems that use Linux as the kernel. The GNU C Library is primarily designed to be a portable and high performance C library. It follows all relevant standards including ISO C23 and POSIX.1-2024. It is also internationalized and has one of the most complete internationalization interfaces known. The GNU C Library website is at http://www.gnu. ... /software/libc/ Packages for the 2.44 release may be downloaded from: http://ftpmirr ... .gnu.org/libc/ http://ftp.gn ... org/gnu/libc/ The mirror list is at http://www.gnu. ... /order/ftp.html Distributions are encouraged to track the release/* branches corresponding to the releases they are using. The release branches will be updated with conservative bug fixes and new features while retaining backwards compatibility. NEWS for version 2.44 ========== Major new features: System-wide tunables can be applied using /etc/tunables.conf and running ldconfig. Specific tunable settings and the /etc/tunables.conf file format and path are not part of the stable library interfaces and may change between

releases. A new tunable, glibc.elf.thp, is added to map read-only segments with Transparent Huge Pages (THP) if THP is not disabled in the kernel. When glibc.elf.thp is set to 1, malloc uses the actual kernel THP mode instead of defaulting to madvise mode and madvise_thp will stop issuing MADV_HUGEPAGE if kernel THP mode is always. The THP page size in malloc is capped to MAX_THP_PAGESIZE. If the THP page size is above MAX_THP_PAGESIZE, THP in malloc is disabled. Additional optimized and correctly rounded mathematical functions have been imported from the CORE-MATH project, in particular cosh, sinh, and tanh. Many additional improvements to existing functions have been synchronized from the CORE-MATH project. For C++26, the assert macro is now variadic, allowing more complex arguments containing commas (which however still must evaluate to a single value). The SVID error handling for cosh and sinh was moved to compatibility symbols, allowing improvements in performance. Static PIE is now supported for arm-*-linux-gnueabi. It requires toolchain support to correctly set the expected linker options. On AArch64 targets that support the Guarded Control Stack extension all GCS operations (including status, write on shadow stack, and push to shadow stack) are locked after enabling GCS with ENFORCED or OVERRIDE GCS policy. When a GCS operation is locked, a program cannot change this operation status via the prctl syscall. This prevents disabling or corrupting the GCS shadow stack during runtime. On AArch64 targets, log, exp, sin, cas, sinh, cosh, asinh, acosh, atanh single and double precision special cases have been vectorized for SVE and AdvSIMD, and vector variants of powr have been added. On RISC-V targets, vector extension optimized variants of memcmp, memccpy, memchr, memcpy, memmove, stpncpy, strcmp, strchr, strcpy, strncmp, strncpy, strlen, and strrchr have been added. On PowerPC, memchr optimized for Power10 has been re-added. Support for LoongArch32 has been added. Pre-built ld.so.cache files can be installed with ldconfig. A new locale has been added: hrx_BR (Hunsrik language spoken in Brazil). Deprecated and removed features, and other changes affecting compatibility: Although malloc and related functions currently return pointers aligned to alignof (max_align_t), the documentation now says future versions of glibc may relax alignment requirements for small allocations. For example, a future malloc(1) might return a pointer with odd alignment, because no object of size 1 can have a fundamental alignment greater than 1. The s390-linux-gnu (31bit) configuration is no longer supported. The --enable-memory-tagging configure option has been removed. The corresponding AArch64-specific functionality that was previously activated by this flag has been removed as well. The --enable-static-nss configure option has been removed. It had no effect on the build since the NSS reorganization in glibc 2.33; its only remaining behavior was to suppress the link-time warnings on the NSS interface functions in libc.a, which are now emitted unconditionally. Security related changes: The following CVEs were fixed in this release, details of which can be found in the advisories directory of the release tarball: GLIBC-SA-2026-0005: gethostbyaddr and gethostbyaddr_r may incorrectly handle DNS response (CVE-2026-4437) GLIBC-SA-2026-0006: gethostbyaddr and gethostbyaddr_r return invalid DNS hostnames (CVE-2026-4438) GLIBC-SA-2026-0007: iconv crash due to assertion failure with untrusted input (CVE-2026-4046) The following bugs were resolved with this release: [2363] libc: EOPNOTSUPP and ENOTSUP in errno.h must be different, according to SUSv3 [3794] manual: iconv: TRANSLIT and IGNORE feature not documented [15792] dynamic-link: [arm] ARM dynamic linker should save/restore coprocessor registers [20331] libc: fts ignores errors from readdir() [20680] dynamic-link: ifunc resolver cannot access the thread pointer with static linking [22944] libc: fts cannot traverse paths which have a length longer than USHRT_MAX [25257] libc: sotruss: fix error message for '--f' argument [25770] locale: newlocale memory leak in LOCPATH parsing and on error paths [27582] libc: x86_64: IFUNC in static user programs may crash when built with -fstack-protector-all [28218] dynamic-link: ld.so: ifunc resolver calls a lazy PLT. When does it work? [28817] libc: static-pie ifunc resolver tls failure [28940] nss: __nss_database_get doesn't check for allocation failure [30136] manual: Please document behaviour of iconv(3) when input is untranslatable [30304] nptl: nptl/tst-pthread-gdb-attach test fails with new libc shared

library version [30769] malloc: malloc_trim is not working correctly for arenas other than arena 0 [30976] dynamic-link: rtld: resolve ifunc relocations after JUMP_SLOT/GLOB_DAT/etc [30992] libc: alpha: setrlimit() with negative values besides RLIM_INFINITY returns EPERM [31901] libc: elf/tst-glibc-hwcaps-prepend-cache fails on i686 [33226] math: math-vector-fortran.h vs not ffast-math [33626] libc: execvpe should skip inaccessible $PATH components [33650] build: abilist.awk doesn't handle unversioned defined symbols [33785] stdio: New streams are linked into global list before they are fully initialized [33848] build: Build fails at openat2.h, redefinition of 'struct open_how' [33882] libc: Recursion in nftw() causes stack overflow(CWE-674) [33904] build: error: '__vasprintf_chk' undeclared here [33921] build: Building with Linux-7.0-rc1 errors on OPEN_TREE_CLONE [33935] stdio: _IO_wfile_doallocate not linked correctly when linking glibc statically [33980] locale: iconv: ibm139x trigger assertion error when converting to internal while lack enough room (CVE-2026-4046) [33985] build: ld: cannot find -lgcc_s: No such file or directory [33999] stdio: libio: potential dangling _IO_save_base or memory leak in wgenops.c [34006] stdio: libio: inconsistent fmemopen_write behavior on last \0 [34008] stdio: stdio-common: scanf %mc pattern will cause heap overflow when width > 1024 [34014] nss: gethostbyaddr and gethostbyaddr_r may incorrectly handle DNS response [34015] nss: gethostbyaddr and gethostbyaddr_r return invalid DNS hostnames [34019] stdio: libio: undefined behavior when setbuf on open_memstream [34033] network: resolv/ns_print.c: ns_sprintrrf TSIG path bypasses buflen and can overflow caller buffer [34064] dynamic-link: The unnecessary PT_NOTE check in when loading a binary [34069] network: Buffer overread in ns_sprintrrf with corrupted RDATA field (CVE-2026-6238) [34070] hurd: Calling open ("/dev/tty/", O_RDONLY) causes the program to segfault [34073] regex: regexec can mistakenly match with backrefs and the $ anchor [34079] dynamic-link: THP segment load aligns all PT_LOAD segments to THP page size [34080] dynamic-link: Support THP segment load with THP enabled with madvise [34083] dynamic-link: __get_thp_mode and __get_thp_size are called twice [34090] libc: wordexp WRDE_APPEND rollback restores stale we_wordv, leading to invalid free in wordfree [34098] libc: Missing SUPPORT_STATIC_PIE in arm32 [34129] string: x86: Non-temporal memset unreachable on AMD Zen 3/4/5 [34144] libc: ld.so clobbers VFP registers during runtime linking [34154] network: Segfault in sock_eq after res_init() returns -1, due to stale _u._ext.nscount in __res_iclose [34156] dynamic-link: dlsym(RTLD_DEFAULT, ...) from a constructor SIGSEGVs when tail-called [34164] dynamic-link: elf: IFUNC resolvers do not see static TLS initialization [34170] dynamic-link: elf: IFUNC resolver reading global- dynamic/TLSDESC __thread variable crashes inside __tls_get_addr [34183] math: fma produces wrong results [34192] nptl: pthread_setname_np opens /proc/<tid>/comm with O_RDWR instead of O_WRONLY|O_CLOEXEC [34196] libc: elf: static dlopen: pointer guard of the loaded ld.so/libc.so is left uninitialized [34197] dynamic-link: elf: Stack canary and pointer guard are recoverable from AT_RANDOM (getauxval) [34205] libc: aarch64: SIGSEGV in tunable_strcmp in static-pie binaries run with a string tunable [34208] stdio: scanf not pushback after matching failure [34210] libc: elf/tst-glibc-hwcaps-prepend-cache fails on armv7a-unknown-linux-gnueabihf [34236] locale: Non-representable transliteration still causes iconv to exit with 1 if TRANSLIT is specified [34289] network: ns_sprintrrf uses p_class, p_type internally [34311] build: THP tests failed to link [34347] libc: Incorrect trailing bitfield word of struct tcp_info [34348] dynamic-link: FAIL: elf/tst-thp-1 if THP is disabled in kernel [34351] build: Random test failures [34355] build: [2.44 Regression] "make check -j7 subdirs=stdio-common" no longer works [34396] libc: sparc64-unknown-linux-gnu , Gentoo: >200 test failures, SIGILL in many binaries [34398] string: Truncated strncpy on s390x z900 ifunc variant Release Notes ========== https://sourcewar ... wiki/Release/2.44 Contributors ========== This release was made possible by the contributions of many people. The maintainers are grateful to everyone who has contributed changes or bug reports. These include: Adam Yi Adhemerval Zanella Alejandro Colomar Andreas K. Hüttel Andreas Schwab Arjun Shankar Aurelien Jarno Avinal Kumar Brian Jorgensen Carlos O'Donell Carlos Peón Costa

Charlotte Mcmenamin Collin Funk Cosmina Dunca DJ Delorie Daan De Meyer Deng Jianbo Dev Jain Diego Nieto Cid Dmitry Kovalenko Dylan Fleming Etienne Brateau Fabian Rast Florian Weimer Frédéric Bérat Garccez George Hu H.J. Lu Jakub Jelinek Jiamei Xie Jiho Lee Jiri Stransky John David Anglin Jonathan Wakely Josef Johansson Joseph Myers Justus Winter Luca Boccassi Lucas Chollet Martin Coufal Matt Turner Michael Ford Michael Jeanson Michael Kelly Mike FABIAN Mike Kelly Muhammad Kamran Nicolas Boulenguez Paul Eggert Peter Bergner Peter Collingbourne Petr Menšík Pierre Blanchard Pino Toscano Pádraig Brady Richard Wild Rocket Ma RyotaSaito Sachin Monga Sajan Karumanchi Sam James Samuel Balazi Samuel Thibault Sana Kazi Sergey Kolosov Shamil Abdulaev Shengwen Cheng Siddhesh Poyarekar Stefan Liebler Thomas Daubney Tomasz Kamiński Uros Bizjak WANG Rui Weihong Ye Weixie Cui Wilco Dijkstra Xi Ruoyao Xiang Gao Yao Zihong Yunze Zhu Yury Khrustalev Zihong Yao mengqinggang xiejiamei zombie12138 We would like to call out the following and thank them for their tireless patch review: Adhemerval Zanella Andreas K. Hüttel Arjun Shankar Aurelien Jarno caiyinyu Carlos O'Donell Collin Funk DJ Delorie Florian Weimer Frédéric Bérat Ganesh Gopalasubramanian H.J. Lu JiangNing Mathieu Desnoyers Paul Eggert Paul Zimmermann Peter Bergner Sam James Samuel Thibault Siddhesh Poyarekar Stefan Liebler Sunil K Pandey Wilco Dijkstra Yury Khrustalev

- health @ Savannah: GNU Health en la facultad de Ciencias Sociales de la Universidad de Buenos Aires (2026/07/23 12:41)
  Los próximos días 5, 6 y 7 de agosto tendrán lugar las XVII Jornadas Nacionales de Debate Interdisciplinario en Salud y Población “Investigar e intervenir en salud en tiempos de negacionismos y retrocesos”, organizadas por el Área de Salud y Población del Instituto de Investigaciones Gino Germani de la Facultad de Ciencias Sociales de la Universidad de Buenos Aires (UBA). Luis Falcón (GNU Solidario) junto al Dr. Fernando Sassetti (UNER) presentarán en la sección "Desigualdades Sociales de la Salud", con el título "Software Libre como modelo de equidad, privacidad, soberanía tecnológica y sostenibilidad en salud. El caso de GNU Health". Para la comunidad de GNU Health es un privilegio y un honor ser parte de este tan importante evento que lucha por la dignidad del individuo y de la comunidad, por un sistema sanitario público, de calidad y universal. Un sistema y un derecho hoy seriamente comprometido y amenazado por las grandes corporaciones financieras y tecnológicas. Haciendo alusión al título de las jornadas, la comunidad GNU y la filosofía del Software Libre representan el faro moral para Investigar e intervenir en salud en tiempos de negacionismos y retrocesos. ¡Nos vemos en Buenos Aires!
- osip @ Savannah: osip2 [5.3.2] (2026/07/22 23:17)
  A new security release was published today! A minor Out-of-bounds Read was discovered. And fixed! No confidential impact is possible. A very low risk of crash is possible. Enjoy & update! Aymeric
- parallel @ Savannah: GNU Parallel 20260722 ('Chat Control') released [stable] (2026/07/21 14:39)
  GNU Parallel 20260722 ('Chat Control') has been released. It is available for download at: lbry://@GnuParallel:4 Quote of the month: gnu parallelすごい！！！ ーー たらたら@nosennyuu@twitter New in this release: Bug fixes and man page updates. 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 ... rg/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.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/1 ... 81/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.

- GNUnet News: libgnunetchat 0.8.0 (2026/07/17 22:00)
  libgnunetchat 0.8.0 released We are pleased to announce the release of libgnunetchat 0.8.0. This is a minor new release bringing compatibility with the major changes in latest GNUnet release 0.28.0. Some minor issues in the API got fixed. Additionally the library was updated to make use of the newer PILS service and an additional layer of encryption for shared files got removed. It is intended to rely on the encryption layer of the FS service in GNUnet for that in the future to reduce overall complexity. Older releases of the applications using libgnunetchat stay compatible with this release. Download links libgnunetchat-0.8.0.tar.gz libgnunetchat-0.8.0.tar.gz.sig 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 http://ftp.gnu.org/gnu/gnunet/ Noteworthy changes in 0.8.0 Remove additional file encryption layer besides FS implicit layer Fix issue creating duplicate contexts/chats for individual contacts A detailed list of changes can be found in the ChangeLog .
- GNUnet News: GNUnet 0.28.0 (2026/07/16 22:00)

GNUnet 0.28.0 released We are pleased to announce the release of GNUnet 0.28.0. 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 0.27.X versions. Please be aware that Git master is thus henceforth (and has been for a while) INCOMPATIBLE with the 0.27.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 0.28.0 release is still only suitable for early adopters with some reasonable pain tolerance . Download links gnunet-0.28.0.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.

- libtool @ Savannah: libtool-2.6.2 released [stable] (2026/07/16 16:47)
  Libtoolers! The Libtool Team is pleased to announce the release of libtool 2.6.2. GNU Libtool hides the complexity of using shared libraries behind a consistent, portable interface. GNU Libtool ships with GNU libltdl, which hides the complexity of loading dynamic runtime libraries (modules) behind a consistent, portable interface. See the NEWS below for a brief summary. Thanks to everyone who has contributed! The following people contributed changes to this release: libtool 2.6.2 [stable]: Ileana Dumitrescu (4) libtool 2.6.1 [beta]: Alexandre Janniaux (4) Alexey Samsonov (1) Anthony Mallet (1) Arnold (1) Dima Pasechnik (1) Frederic Berat (1) Ileana Dumitrescu (15) KO Myung-Hun (4) Kirill Makurin (1) Mintsuki (1) Nicolas Boulenguez (1) Olly Betts (1) Patrice Dumas (1) Richard J. Mathar (1) libtool 2.6.0 [alpha]: Anthony Mallet (1) Bruno Haible (2) Christian Feld (1) Collin Funk (1) Elizabeth Figura (1) Evgeny Grin (1) Frédéric Bérat (1) Gleb Popov (1) Ileana Dumitrescu (47) Julien ÉLIE (1) Karl Berry (1) Kirill Makurin (1) Manoj Gupta (1) Martin Storsjö (1) Michael Haubenwallner (2) Mintsuki (1) Mitch (1) Pierre Ossman (2) Takashi Yano (1) Ileana [on behalf of the libtool maintainers] ========== Here is the GNU libtool home page: https://gnu. ... g/s/libtool/ Here are the compressed sources: https://ftpmirror ... tool-2.6.2.tar.gz (2.1MB) https://ftpmirror ... tool-2.6.2.tar.xz (1.1MB) Here are the GPG detached signatures: https://ftpmirror ... -2.6.2.tar.gz.sig https://ftpmirror ... -2.6.2.tar.xz.sig Use a mirror for higher download bandwidth: https://www.gnu.o ... rg/order/ftp.html Here are the SHA256 and SHA3-256 checksums: File: libtool-2.6.2.tar.gz SHA256 sum: 24adb3aa9ae035c70faba344af57d73215eb89281045af6c7ccd307751f8b0bf SHA3-256 sum: b0e77c0dc9a082830c95d182da77747d1f5435a06132feefd5054bfde2c9da81 File: libtool-2.6.2.tar.xz SHA256 sum: 2ef1067c16c97db930fd740cc9bc3d3ba9a583804ae5ac42cc3e8719e49e191e SHA3-256 sum:

c24b9995af8391a310a258dcb98897f92d86f47acca235f3d2859ca0ed1d9dd0 Verify the SHA256 checksum with either sha256sum, sha256, or 'shasum -a 256'. Verify the SHA3-256 checksum with 'cksum -a sha3 -l 256 --base64' 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 libtool-2.6.2.tar.gz.sig The signature should match the fingerprint of the following key: pub rsa4096 2021-09-23 [SC] FA26 CA78 4BE1 8892 7F22 B99F 6570 EA01 146F 7354 uid Ileana Dumitrescu <ileanadumitrescu95@gmail.com> uid Ileana Dumitrescu <ileanadumi95@protonmail.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 ileanadumitrescu95@gmail.com gpg --recv-keys 6570EA01146F7354 wget -q -O- 'https://savannah. ... ol&download=1' | gpg --import - As a last resort to find the key, you can try the official GNU keyring: wget -q https://ftp.gnu.o ... u/gnu-keyring.gpg gpg --keyring gnu-keyring.gpg --verify libtool-2.6.2.tar.gz.sig This release is based on the libtool git repository, available as git clone https://https.git ... g/git/libtool.git with commit 309bb53a8adfb22c6e5869cc8da049bf123e5438 tagged as v2.6.2. For a summary of changes and contributors, see: https://gitweb.gi ... shortlog;h=v2.6.2 or run this command from a git-cloned libtool directory: git shortlog v2.6.1..v2.6.2 This release was bootstrapped with the following tools: Autoconf 2.73 Automake 1.18.1 Gnulib 2026-07-03 491f1bb7d3049f3ab7825f3ee665c209658e9965 NEWS Noteworthy changes in release 2.6.2 (2026-07-16) [stable] Please see beta release 2.6.1 and alpha release 2.6.0 for a list of release changes. Noteworthy changes in release 2.6.1 (2026-06-04) [beta] ** New features: - Pass 'resource-dir=*' flag for Clang. - Recognise explicit shared library arguments when linking dependency libraries to a shared library, like exists when linking a program. - Support OpenMP with macOS clang by processing '-Xpreprocessor -fopenmp' as one token. ** Bug fixes: - Store cygpath file path conversions correctly for MSYS2 and MSVC. - Fix syntax error in LT_PROG_OBJC and LT_PROG_OBJCXX. - Separate Objective C and C++ cache check for proper tagging support. - Fix in darwin to support values with spaces. - Limit the length of DLL name to 8.3 correctly to avoid corrupting a generated DLL on OS/2. - Remove unused variable on OS/2, which could cause issues with static library generation if defined. - Recognise more static linking options for Clang. - Fix emscripten CXX postdeps using non-PIC sysroot. - Avoid deprecated option '-o' with MSVC compilers and replace with '-Fe'. - Avoid overlinking of dependency libraries on ELF systems. - Ensure old libraries are not archived. ** Changes in supported systems or compilers: - Add support for SlimCC compiler. - Add support for *-ironclad-gnu. Noteworthy changes in release 2.6.0 (2025-09-18) [alpha] ** New features: - Add a new tool, libtool-next-version, to guide users through updating library versions. - Add tagging for Objective-C and Objective-C++, OBJC and OBJCXX. - Increase 5 digit limit on revision value for libraries to 19 digits, which is referencing Unix epoch time in nanoseconds. - Add configuration options to choose whether to use '-nostdlib' to let the compiler frontend decide what standard libraries to link when building C++ shared libraries and modules, --enable-cxx-stdlib and --disable-cxx-stdlib. - Allow statically linking GCC and Clang compiler support libraries into shared libraries. - Add linking clang_rt static archives compiler internal libraries by their absolute path. - Set 'mklink' as the symlinking tool for MSVC. - Pass '--target' architecture flag for Clang. - Support MSYS and MSYS2 file path conversions. ** Bug fixes: - Fix wrongly deduplicated compiler dependencies on linux. - Fix NetBSD postdeps for shared libraries. - Fix statically linking dependencies into shared C++ libraries when utilizing clang builtins or g++ options like, -static-libstdc++, by using a new configuration option, --enable-cxx-stdlib. - Ensure *-linux-mlibc host matches to mlibc userland rather than matching to GNU/Linux and similar userlands. - Fix hang with cmd.exe in MSYS. - For MSVC, fix mishandling compiler flags, symlinking, cl.exe '.exp' extension collision, symbol names, and numerous testsuite bugs. - Fix undeclared reference to access on Windows in libltdl. - Fix flang -Wl flags on FreeBSD. - Fix reordering '--as-needed' flag.

- Fix libltdl early failures for multi-arch. ** Changes in supported systems or compilers: - Support additional Intel OneAPI compilers, 'icx', 'icpx', and 'ifx'. - Support ML64 (Microsoft Macro Assembler). Enjoy!
- GNU Taler news: Taler lecture at Cedarcrypt 2026 (2026/07/15 22:00)
  Özgür Kesim from the GNU Taler team will give a lecture about Taler, the protocol suite, at Cedarcrypt 2026, a new applied cryptography summer school and conference, July 13-16 in Paphos, Cyprus.
- FSF Events: Free Software Directory meeting on IRC: Friday, July 24, starting at 12:00 EDT (16:00 UTC) (2026/07/15 14:10)
  Join the FSF and friends on Friday, July 24 from 12:00 to 15:00 EDT (16:00 to 19:00 UTC) to help improve the Free Software Directory.
- FSF Events: LibreLocal meetup in Noor, Iran (2026/07/15 13:55)
  July 30, 2026 from 17:00â€“19:30 IRST.
- screen @ Savannah: GNU Screen v.5.0.2 is released (2026/07/12 22:36)
  Hi everyone, I'm glad to announce the new release of GNU screen. Screen is a full-screen window manager that multiplexes a physical terminal between several processes, typically interactive shells. The 5.0.2 release includes the following changes to the previous release 5.0.1: Add %* escape to output caret character literal ("^") Add portable PAM conversation callback prototype (for Solaris) Fixes: - type on big-endian systems - UTF-8 combining sequences - buffer overflow in SendCmdMessage() - detaching fail with empty terminfo and leave the session attached - eliminates the TOCTOU race - manpage fixes Release (official tarball) will be available soon for download: https://ftp.gn ... rg/gnu/screen/ Please report any bugs or regressions. Thanks to everyone who contributed to this release. Cheers, Alex
- findutils @ Savannah: GNU findutils 4.11.0 released (2026/07/11 09:52)
  This is to announce findutils-4.11.0, a stable release. This release follows the recent POSIX (IEEE Std 1003.1-2024) changes, especially to mention the new behavior of 'find -mount' vs. 'find -xdev', as well as a lot of documentation improvements. See the NEWS below for more details. GNU findutils is a set of software tools for finding files that match certain criteria and for performing various operations on them. Findutils includes the programs "find", "xargs" and "locate". More information about findutils is available at: https://www.gnu.o ... ftware/findutils/ Please report bugs and problems with this release via the the GNU Savannah bug tracker: https://savannah. ... /?group=findutils Please send general comments and feedback about the GNU findutils package to the mailing list (<mailto:bug-findutils@gnu.org): https://lists.gnu ... nfo/bug-findutils There have been 186 commits by 10 people in the - sigh - 25 months since 4.10.0: Bernhard Voelker (77) Bjarni Ingi Gislason (1) Christoph Anton Mitterer (1) Collin Funk (5) Dave (1) G. Branden Robinson (42) James Youngman (55) Luk303241305241 Zaoral (1) danny mcClanahan (1) raf (2) This release was bootstrapped with the following tools: Autoconf 2.72 Automake 1.17 M4 1.4.19 Gnulib v1.0-3131-ga575239e47 Please consider supporting the Free Software Foundation in its fund raising appeal; see <https://www. ... .org/appeal/>. Thanks to everyone who has contributed! Have a nice day, Bernhard Voelker & James Youngman [on behalf of the GNU findutils maintainers]
  ========== Here are the compressed sources: https://ftp.gnu.o ... ils-4.11.0.tar.xz Here are the GPG detached signatures[*]: https://ftp.gnu.o ... 4.11.0.tar.xz.sig Use a mirror for higher download bandwidth: http://www.gnu. ... /order/ftp.html Here is the SHA256 checksum: bfd19cb06cc71f3352d567e90284d8cdac02ac89774bbeadf0b533b0c11432fd findutils-4.11.0.tar.xz [*] 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 findutils-4.11.0.tar.xz.sig If that command fails because you don't have the required public key, then run this command to import it: gpg --keyserver keys.gnupg.net --recv-keys 0CF4E8D871593224842832B888DD9E08C5DDACB9 and rerun the 'gpg --verify' command.
========== NEWS Noteworthy changes in release 4.11.0 (2026-07-11) [stable] ** Bug Fixes find no longer crashes when diagnosing a directory cycle (without a symlink being involved pointing to a parent directory). [Bug present since the FTS implementation.] 'find -used' now behaves correctly on OpenBSD 7.8 with difftime(3) underflow bug in the C library (already fixed there) when the access time of a file is identical to its change time. [#68264] 'find -ignore_readdir_race' now better handles races between FTS reading a directory and visiting its entries when the file or directory was meanwhile removed. [#45930] To fix a POSIX compatibility bug, -exec foo Z{} + is no longer a complete predicate, because '+' is only a terminator when it follows an argument which is exactly '{}'. The findutils documentation already states this, and now find's behaviour matches the documentation. [#66365] 'updatedb.sh' now properly handles the variables for the 'find' and 'frcode' utilities, and hence avoids command injection. ** Changes in find As announced since the release of 4.7.0 (2019) and mandated by POSIX 2024, the behaviour of the -mount option changed: while it was a mere alias for the -xdev option to prevent descending into directories of another device, the -mount option now makes find(1) ignore files on another device, i.e., 'find -mount' will skip the entry of active mount points already. Example, assuming the PROC filesystem is mounted on '/proc': $ find / -mount -path /proc -print $ find / -xdev -path /proc -print /proc [#54745] The actions -execdir and -okdir now refuse the '{}' replacement in the zeroth argument of the command to be run. While POSIX allows this for -exec, this is deemed insecure as an attacker could influence which files could be found. 'find -regex' with the default or the 'emacs' regextype now aligns better with Emacs behaviour, and therefore e.g. supports character classes: $ touch 123 && find -regex './12[[:digit:]]' ./123 find now issues a warning when the punctuation operators '(', ')', '!' and ',' are passed with a leading dash, e.g. '-!'. Future releases will not accept that any more. Accepting that was rather a bug "since the beginning". ** Improvements xargs now gives a better error diagnostic when executing the given command failed. ** Documentation Changes The most recent version of the POSIX standard (IEEE Std 1003.1-2024, also known as The Open Group Base Specifications, Issue 8) has standardised "find -print0" and "xargs -0". Our documentation now points this out. Similarly for 'find -iname'. The code example for "Finding the Shallowest Instance" in the Texinfo manual and the corresponding one in the EXAMPLES section in the find.1 man page have been fixed. [#62259] Translators contributed numerous fixes for issues in the find.1 man page. The list of actions that suppress the default -print action has been supplemented with the missing '-print0' and '-fprint0' actions. The manual pages have been updated to give better and/or more consistent output with manpage formatters other than GNU roff. ** Translations Updated the following translations: Arabic, Brazilian Portuguese, Bulgarian, Chinese (simplified), Croatian, Czech, Dutch, Estonian, French, German, Indonesian, Korean, Polish, Portuguese, Romanian, Spanish, Swedish, Ukrainian. ** Future Changes A future release will remove the warning message find prints about the 2007 change in the meaning of "-perm /000". Everybody who is likely to care probably knows about this change by now. -eof-

- FSF Events: Free Software Directory meeting on IRC: Friday, July 17, starting at 12:00 EDT (16:00 UTC) (2026/07/10 14:31)
  Join the FSF and friends on Friday, July 17 from 12:00 to 15:00 EDT (16:00 to 19:00 UTC) to help improve the Free Software Directory.
- FSF Events: Free Software Directory meeting on IRC: Friday, July 10, starting at 12:00 EDT (16:00 UTC) (2026/07/06 20:45)
  Join the FSF and friends on Friday, July 10 from 12:00 to 15:00 EDT (16:00 to 19:00 UTC) to help improve the Free Software Directory.

- cssc @ Savannah: CSSC-1.5.0 released (2026/07/05 21:38)
  This is to announce CSSC-1.5.0, a beta release. There have been 424 commits by 2 people since 1.4.1. Thanks to Greg A. Woods for helping to improve CSSC. See the NEWS below for a brief summary. ========== Here is the GNU CSSC home page: https://gn ... rg/s/CSSC/ Here are the compressed sources and a GPG detached signature: https://alpha.gnu ... CSSC-1.5.0.tar.gz https://alpha.gnu ... -1.5.0.tar.gz.sig Use a mirror for higher download bandwidth: https://www.gnu.o ... rg/order/ftp.html Here are the SHA256 and SHA3-256 checksums: File: CSSC-1.5.0.tar.gz SHA256 sum: 8483a31aac756955843ef574bed6fa9e052ea492217f3882033487d2704c6cf4 SHA3-256 sum: c138c32cab373a51c32d2049064532ac24a6fa777b7edecabe4d605460018845 Verify the SHA256 checksum with either sha256sum, sha256, or 'shasum -a 256'. Verify the SHA3-256 checksum with 'cksum -a sha3 -l 256 --base64' 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 CSSC-1.5.0.tar.gz.sig The signature should match the fingerprint of the following key: pub rsa4096 2015-12-24 [SC] 0CF4 E8D8 7159 3224 8428 32B8 88DD 9E08 C5DD ACB9 uid James Youngman <james@youngman.org> uid James Youngman <jay@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 james@youngman.org gpg --recv-keys 88DD9E08C5DDACB9 wget -q -O- 'https://savannah. ... SC&download=1' | gpg --import - As a last resort to find the key, you can try the official GNU keyring: wget -q https://ftp.gnu.o ... u/gnu-keyring.gpg gpg --keyring gnu-keyring.gpg --verify CSSC-1.5.0.tar.gz.sig This release is based on the CSSC git repository, available as git clone https://https.git ... .org/git/CSSC.git with commit 77bdce39a09b9ff37831c4dd1cb0d4dd5967cb39 tagged as v1.5.0. For a summary of changes and contributors, see: https://gitweb.gi ... shortlog;h=v1.5.0 or run this command from a git-cloned CSSC directory: git shortlog CSSC-1.4.1..v1.5.0 This release was bootstrapped with the following tools: Autoconf 2.72 Automake 1.17 Gnulib 2026-06-08 88592a2880cf39a2f597cd0294a90d8dd7faa2df NEWS Noteworthy changes in release 1.5.0 (2026-07-05) [beta] * The test suite no longer depends on Python, so it should work on older systems that have no Python 3 interpreter. While you can build and test CSSC without Python, you do need a Python interpreter to do some maintenance tasks (such as importing the "gnulib" code into a git checkout). * Tolerate SCCS files in which file flags lack the space separator which would normally come between the flag letter and the associated value. Noteworthy changes in release 1.5.0-rc3 (2026-06-14) * Some typos in error message have been fixed. * admin now supports combination of -r with -n as well as the portable combination of -r with -i. * Support "sccs sact"; the sact program already existed but could not previously be invoked via the sccs wrapper. Thanks to Greg A. Woods for this improvement. * In some places we now prefer "grep -E" to "egrep" in order to avoid a warning message from GNU grep. Some very old versions of Unix may not support this option. * Various C++ portability improvements. * Updated version of gnulib. * Updated version of googletest; this is now at the last version at which it still supported building with Automake. Noteworthy changes in release 1.5.0-rc2 (2024-05-13) * This release is more careful to detect I/O failures when writing to stdout in prs and prt. * This release should build on more modern platforms. * The --with-googletest configure option is removed (now we always use it). * Updated versions of gnulib and googletest.

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**