The PCLinuxOS magazine

Volume 234

July, 2026

ICYMI: California Exempts Linux, Open Source OS's From Age Verification Law

Wiki Pick: Getting Rid Of Unwanted/Unneeded Files

GIMP Tutorial: The Resynthesizer Plugin

A New Script To Create Image Transparency

Tip Top Tips: KDE 6.x.y Change Shutdown/Logout Countdown (Konsole Method)

A Custom Script To Manage Your Wireguard VPN Connection

PCLinuxOS Recipe Corner: Chicken Lazone

Alternative "Fixes" For The NLUUG Repo Path Problem

PCLinuxOS Puzzled Partitions

And more inside...

In This Issue...

The PCLinuxOS magazine

The PCLinuxOS name, logo and colors are the trademark of Texstar. **The PCLinuxOS Magazine** is a monthly online publication containing PCLinuxOS-related materials. It is published primarily for members of the PCLinuxOS community. The magazine staff is comprised of volunteers from the PCLinuxOS community.

Visit us online at https://pclosmag.com.

This release was made possible by the following volunteers:
Chief Editor: Paul Arnote (parnote)
Assistant Editor: Meemaw
Artwork: Paul Arnote, Meemaw
PDF Layout: Paul Arnote, Meemaw
HTML Layout: tbs, horusfalcon

Staff:
YouCanToo David Pardue
Alessandro Ebersol

Contributors:
Cuig keltonix

From The Chief Editor's Desk

Seventeen years ago, in July 2009, I was named the Chief Editor of The PCLinuxOS Magazine. It's a position that I've held ever since.

I took over after one of the largest upheavals in PCLinuxOS history. The then editor of The PCLinuxOS Magazine, essentially, tried to hold the magazine hostage in an effort to kill the magazine.

For about a year prior to July 2009, the magazine would not appear on a regular, monthly basis. The then editor said that there wasn't enough to write about. During a sabbatical by Texstar for health reasons, a division of allegiance emerged, with some of the developers threatening to take over PCLinuxOS. The then editor "sided" with that faction that attempted their coup of PCLinuxOS.

That forced Texstar to return from his sabbatical earlier than he planned to reassert his control over the distro that he had created. In the aftermath, many members of that attempted coup fled, vowing to make their own distro (which, by any and all reasonable metrics, failed). The previous editor of the magazine went with them after Texstar essentially fired him. It's needless to say that trust in the magazine was at an all-time low.

Back then, we had another forum where development ideas were put forth and tried out. It was called MyPCLinuxOS. Our former and

departed moderator Archie Arevalo put out a call to resurrect the magazine, which, at that time, had been published on a mostly monthly basis (with the exception of the tenure of the former editor). Having a journalism background (I had worked as a photojournalist for a number of years before then, both for local daily newspapers and for the wire services), I immediately threw my name into the hat to be the person to lead that resurrection of the magazine.

Initially, Archie told me that this "job" would last only a year or two, and then I would "hand

it over" to someone else. Except, no one else emerged to take over. Our first assistant editor, Andrew Strick, was in law school, and is now employed as an Attorney Editor for Thomson Reuters. He was (rightfully) focused on his education and career, and didn't stick around too long. Meemaw also joined me soon after me taking on the job as the magazine's chief editor as the magazine's assistant editor (for a while, we listed both Andrew and Meemaw as assistant editors), but she made it clear that she had no interest in taking over as the chief editor. Over time, we worked to reestablish Texstar's trust in the magazine.

And, here I am, 17 years later, still pumping out a new issue of The PCLinuxOS Magazine every month. In the time that I've been the chief editor, we've ***never*** missed putting out a monthly issue. And, to put the cherry on top of the cake, we've also put out several special editions of the magazine. So much for there not being enough to write about, huh?

In the beginning, we tapped into the real, everyday talents of other forum members, such as ms_meme, georgetoon, YouCanToo, and our departed friend tuxlink. We continue to do that to this day. We brainstormed and came up with numerous regular and recurring columns, as well as tapping into anything that looked like it might create a nice article that PCLinuxOS users might be interested in. Most of all, we've made a concerted effort to ensure that every issue of the magazine includes something for as many users as possible, from rank beginners to more experienced Linux users.

As with any Linux distro, we've seen our fair share of people and characters come and go. We've seen some of our forum members and magazine supporters succumb to health problems and criminal actions. Yet, through it all, we never missed putting out a monthly issue.

Having started in September 2006, The PCLinuxOS Magazine is the oldest community based Linux magazine in existence. Certainly, we've run our fair share of reprint articles from the open source community, but we've never missed putting out a monthly issue (at least during my time at the helm). Some of our "competitors" have released issues with blank pages in them because a promised article never materialized. We've experienced the same before, but always found things to fill the pages of the magazine. If you just look around, there's ample topics to fill the pages of a community based Linux magazine, and we've proven that resoundingly.

Over my tenure as the magazine's chief editor, I've personally seen a lot of things happen. First, my son Ryan was born, followed three years later by my daughter, Lexi (it's the nickname she goes by … her actual name is Alexandra). My kids have started school, became involved in Cub Scouts and Scouting, and we've started karate classes as a family. I've retired from a 35-year career as a respiratory therapist working in the hospital. One of our forum moderators was killed by his estranged soon-to-be ex-wife. Another lost his life to cancer. Archie appears to have succumbed to the injuries he suffered in an automobile accident. The list of those we've lost goes on for a good bit. Those I've listed are just a sampling of that list. But still, we forge ahead with the same mission that we started with 17 years ago.

Even 17 years later, I actually still love the "job" of serving as the chief editor of The PCLinuxOS Magazine. Although I don't have a "crystal ball," I don't foresee me giving up this position any time soon. After all of this time, it has become just another thing that I do.

In another way of looking at it, I started with issue 30 as the chief editor. Including this issue, we're up to issue 234. Add in another 15 special editions of The PCLinuxOS Magazine that we've published, and we're up to 219 issues under my watch … and still counting. And I can't even begin to count the number of articles I've written over the past 17 years!

So, not only have I been heading up The PCLinuxOS Magazine for the past 17 years, but 2026 also marks the 20th anniversary of the magazine. So here's to many, many, many more years to come!

****

This month's cover celebrates the 2026 FIFA World Cup, showing Tux dressed in a soccer jersey, holding a soccer ball (football to the "rest of the world"), in front of the flags of the nations participating in this year's tournament. If you're not too familiar with the quadrennial FIFA World Cup, check out this guide. To see when your favorite team might be playing, check out this listing of matches. Some of the

matches are being played about five miles (8 Km) away from my house, at Arrowhead Stadium in Kansas City (don't worry … I won't be able to attend any matches … the tickets are obscenely priced). This image was created with Bing Image Creator, using the following "creation" criteria: "Linux mascot Tux dressed in a soccer jersey, holding a soccer ball, surrounded by the flags of the teams participating in the 2026 FIFA World Cup."

****

Until next month, I bid you peace, happiness, serenity, prosperity, and continued good health!

Like Us On Facebook!
The PCLinuxOS Magazine
PCLinuxOS Fan Club

Screenshot Showcase

Posted by franciscoinblack, on June 5, 2026, running icewm.

ICYMI: California Exempts Linux, Open Source OS's From Age Verification Law

by Paul Arnote (parnote)

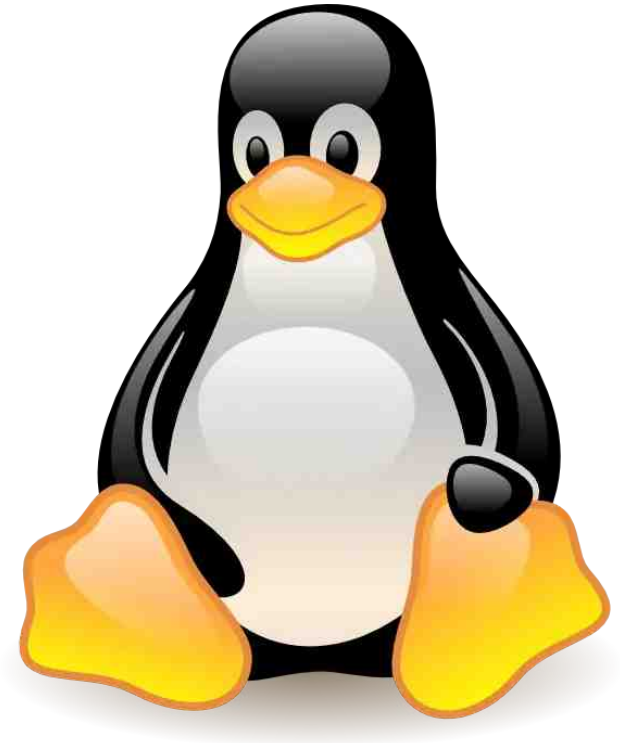
Grapevine PD

Tesla CEO Elon Musk says a lot of silly things. For example, he said that Tesla's steel-plated, but not fully steel-ball-resistant Cybertruck is "apocalypse-level safe." He also said that **"Cybertruck will be waterproof enough to serve briefly as a boat, so it can cross rivers, lakes & even seas that aren't too choppy,"** according to an article from Mashable. Well, one guy apparently took that seriously, and tested it out with his Cybertruck in Grapevine Lake, an 8,000-acre lake northwest of Dallas, Texas. Now, he's in jail. According to Grapevine Police's official account on Facebook, "The driver stated he intentionally drove into the lake to use the Cybertruck's "Wade Mode" feature. The vehicle became disabled and took on water. The driver and passengers abandoned the vehicle and the Grapevine Fire Department Water Rescue Team assisted in removing it from the lake. The driver was arrested on charges of Operation of Vehicle in Closed Section of Park/Lake and numerous water safety equipment violations."

University of Arizona graduates booed former Google CEO Eric Schmidt after he urged them to embrace an AI-shaped future, according to an article from eWeek. Schmidt's commencement address Friday quickly turned tense as his speech turned to AI, job fears, and the pressure facing students entering a changing labor market. According to The Verge, Schmidt acknowledged the anxieties in the room, including fears that "the machines are coming," jobs are disappearing, the climate is breaking, and politics are fractured. He called those fears rational, then pushed graduates toward optimism: *"When someone offers you a seat on the rocketship, you do not ask which seat, you just get on."* That landed badly. For graduates entering a labor market already being reshaped by AI, the advice sounded less like reassurance than pressure to accept disruption on someone else's terms.

Researchers at Loma Linda University Health report that eating eggs may be linked to a lower risk of developing Alzheimer's disease in adults age 65 and older, according to an article from ScienceDaily. Their findings suggest that regular egg consumption could play a role in supporting long-term brain health. The study found that people who ate at least one egg per day for five or more days each week had up to a 27% lower risk of being diagnosed with Alzheimer's disease. "Compared to never eating eggs, eating at least five eggs per week can decrease risk of Alzheimer's," said Joan Sabaté, MD, DrPH, a professor at Loma Linda University School of Public Health and the study's principal investigator. Even smaller amounts of egg consumption were associated with benefits. Eating eggs just 1 to 3 times per month was linked to a 17% reduction in risk, while those who ate eggs 2 to 4 times per week saw about a 20% lower risk, Sabaté said. The research, titled Egg intake and the incidence of Alzheimer's disease in the Adventist Health Study-2 cohort linked with Medicare data, was published in the Journal of Nutrition. Scientists conducted the study to better understand how diet, a factor people can change, might influence the likelihood of developing Alzheimer's disease.

ICYMI: California Exempts Linux, Open Source OS's From Age Verification Law



California plans to exclude Linux and most other open-source operating systems from its new age verification law, which takes effect on Jan. 1, 2027, according to an article from Extreme Tech. The change follows massive pushback from the open-source software community. In October 2025, Governor Gavin Newsom signed AB 1043, a new law for OS providers in California. This law requires an OS to collect users' ages or birth dates when they set up their accounts. OS providers then must share this information with app developers through a real-time API. The law divides users into four age groups: under 13, 13-16, 16-18, and 18 and over. Currently, it defines "operating system provider" as anyone who develops, licenses, or maintains operating system software. This includes companies that make Windows, macOS, Android, iOS, Linux distributions, and Valve's SteamOS. But the Linux community is overwhelmingly against applying the law to open-source operating systems. Critics have said it would be very hard to enforce rules for community-run distributions such as Ubuntu and Debian. These systems don't have a central account system and let users download ISO files from various mirrors around the world. Something similar has also happened in Colorado, where Colorado's path here involved some direct community legwork. Carl Richell, the founder of System76, spent some considerable time working with Senator Matt Ball, one of SB26-051's co-authors, to get open source exclusions written into the bill, according to another article from It's Foss.

Linus Torvalds made a post to the Linux Kernel Mailing List (LKML) on May 24, regarding Linux Kernel 7.1-rc5: *To the surprise of absolutely nobody by now, rc5 is pretty big. Quite a bit bigger than rc5's have traditionally been. I'm not entirely happy about it - most of this is totally trivial stuff to random drivers, which obviously makes it all less scary, but at the same time I'm really not convinced the churn is worth it at rc5 time. These things are "fixes", sure, but at the same time a lot of them are simply so irrelevant that I think they'd be better off in a linux-next tree and get merged during the merge window. So I think I'll start being a bit more hardnosed about this kind of unnecessary churn this late in the game. We are supposed to look for *regressions*. Non-critical fixes to long-standing issues are simply not appropriate for this late in the release cycle. End result: this is too big, and this is the heads-up that I'll be pushing back on pointless pull requests with fixes that just aren't that important. And yes, several of these series were triggered by AI code review. Because fixes or not - and trivial or not - these kinds of large rc weeks are *not* conducive to long-term stability. Trivial fixes may be trivial, and have a pretty low chance of causing problems, but "low chance" is still not "zero chance". So people: start looking closer at your pull requests, and ask yourself: "Is this really a regression or serious enough that it shouldn't just go into the development pile?".*

Cybersecurity researchers have disclosed details of a vulnerability in the Linux kernel that remained undetected for nine years, according to an article from The Hacker News. The vulnerability, tracked as CVE-2026-46333 (CVSS score: 5.5), is a case of improper privilege management that could permit an unprivileged local user to disclose sensitive files and execute arbitrary commands as root on default installations of several major distributions like Debian, Fedora, and Ubuntu. It's also codenamed ssh-keysign-pwn. According to Qualys, which discovered the flaw, the problem is rooted in the kernel's __ptrace_may_access() function and was introduced in November 2016. "The primitive is reliable and turns any local shell into a path to root or to sensitive credential material," Saeed Abbasi, senior manager of Threat Research Unit at Qualys, said. Successful exploitation of the flaw could permit a local attacker to disclose /etc/shadow and host private keys under /etc/ssh/*_key, as well as execute arbitrary commands as

root through four different exploits targeting chage, ssh-keysign, pkexec, and accounts-daemon.

Image by Shafin Al Asad Protic from Pixabay

A Chinese cyber-espionage campaign has been targeting telecommunications providers with newly discovered Linux and Windows malware dubbed Showboat and JFMBackdoor, respectively, according to an article from BleepingComputer. The operation has been active since at least mid-2022 and targeted organizations across the Asia Pacific and parts of the Middle East. It was attributed to the Calypso threat group, also tracked as Red Lamassu. According to researchers at Lumen's Black Lotus Labs and PwC Threat Intelligence, the threat actor set up and used multiple telecom-themed domains to impersonate their targets. The Linux implant Calypso uses in these attacks, dubbed Showboat/kworker, is a modular post-exploitation framework built for long-term persistence after initial compromise. The initial infection vector is unknown. According to a report today from Black Lotus Labs, once Showboat is deployed on a target system, it starts collecting information about the host and sends it to a command-and-control (C2) server. The malware can also upload or download files, hide its own process, and establish persistence via a new service. “One notable feature is the 'hide' command, which enables a process to conceal itself on a host machine by retrieving code stored on external websites such as Pastebin or online forums for use as a ‘dead drop’, Lumen's Black Lotus Labs researchers explain.

In a small, preliminary study, an experimental gene-editing treatment dramatically lowered cholesterol levels, perhaps permanently, after just one infusion, scientists reported, according to an article from the New York Times. If confirmed in larger studies, researchers hope the findings may lead to a one-and-done way to prevent heart disease in large numbers of people. Most gene therapies target rare diseases, but cardiovascular disease kills nearly 800,000 Americans a year. “We have these debates and new guidelines that we should be treating people earlier,” said Dr. John H. P. Alexander, a cardiologist at Duke University who was not involved with the study. “A curative therapy would change the game.” The study, published in The New England Journal of Medicine, was an interim analysis of 35 patients in a trial that will involve as many as 85 participants. All have genetically high levels of LDL cholesterol — the bad kind — or heart disease. In the 35 patients, a single infusion of the highest dose of the treatment reduced LDL cholesterol levels by as much as 62 percent. The change has been sustained in a subgroup whose members were treated 18 months ago. It will be followed by a larger study of 200 patients.

If you're concerned that you act too aggressively at times, you might be able to regain your calm. **According to research, there's one supplement that can help reduce your aggression, and by a significant amount**, according to an article from ScienceAlert. The solution, evidence suggests, is to add some omega-3 to your diet. The fatty acids, available as dietary supplements via fish oil capsules and thought to help with mental and physical well-being, could help cut down on aggression, according to a 2024 study.

Image by Satheesh Sankaran from Pixabay

Researchers at the National Institutes of Health (NIH) have uncovered new details about how GLP-1 weight loss drugs such as semaglutide affect brain cells, revealing internal signaling processes that scientists have only begun to understand, according to an article from Science Daily. The findings,

based on experiments in mice, shed light on why these medications work differently from person to person and why their effects often slow down over time. GLP-1 receptor agonists, including drugs like Ozempic and Wegovy, are already known to help reduce appetite and promote weight loss. Scientists have also identified the brain regions involved in those effects. Until now, however, much less was known about what happens inside the neurons targeted by these drugs. Their experiments showed that semaglutide's impact depended heavily on increased levels of cyclic adenosine monophosphate, or cAMP, in the area postrema, a part of the brain involved in appetite regulation. However, the response was not the same in every neuron.

An unusual collection of stars may represent the remnants of a dwarf galaxy that the Milky Way devoured about 10 billion years ago, according to an article from CNN. Astronomers have dubbed the ancient galaxy Loki, after the Norse god of mischief. The finding could change the current understanding of how the Milky Way evolved in the distant past. The vast Milky Way spans about 100,000 light-years and contains anywhere between 100 billion and 400 billion stars, according to NASA. A light-year is the distance light travels in one year, which is 5.88 trillion miles (9.46 trillion kilometers). Our home galaxy wasn't always such a cosmic giant. It grew over time starting about 12 billion years ago by merging with a multitude of dwarf galaxies. But the original size and mass of the Milky Way remain an open question — driving scientists to search for evidence of the galaxies it consumed to determine its history and evolution.

One of Android's most relied-upon apps is suddenly struggling to do its most fundamental job: displaying emails, according to an article from TechRepublic. Several reports have now emerged of Gmail users on some Android devices experiencing flickering screens, disappearing messages, or buggy rendering of their email apps. Complaints appear concentrated on tablets and foldables, particularly Samsung models, though some reports suggest the issue is broader than a single product line. Analysis so far points to a display or rendering issue likely related to Android System WebView, particularly since temporary fixes such as screen rotation appear to force the interface to render correctly. That pattern suggests the issue is affecting content displays rather than content delivery. And while the issue has reportedly been escalated to Google, no permanent fix has been made available.

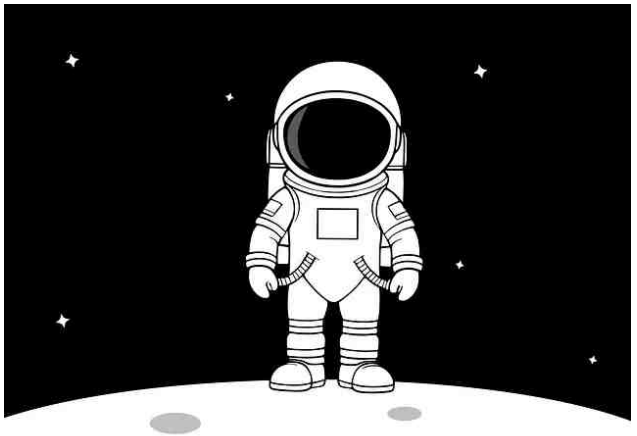
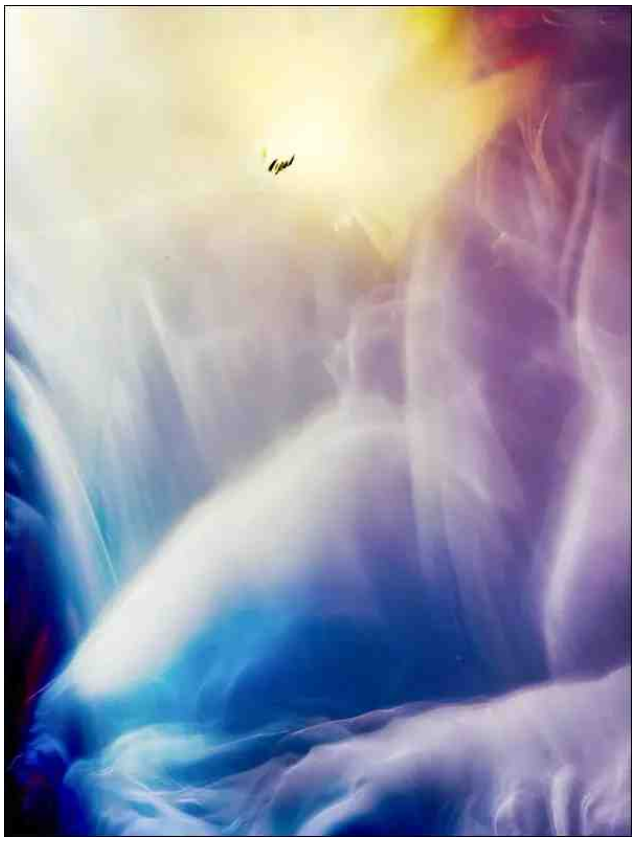


Image by Mohamed Hassan from Pixabay

The first builders of NASA's future Moon base may not need spacesuits, according to an article from eWeek. NASA put its lunar ambitions on full display in Houston this week, using a major student robotics competition to spotlight how machines will help build its future Moon Base. At the 2026 FIRST Robotics World Championship, the agency highlighted its long-term plan for a permanent presence on the Moon while engaging tens of thousands of students, mentors, and parents through exhibits and live demonstrations. NASA's Moon Base concept is designed as a permanent lunar outpost to support science, exploration, and future missions deeper into the solar system, including to Mars. Before astronauts arrive, a first phase of robotic and uncrewed missions will prepare the surface. This includes a rapid series of CLPS (Commercial Lunar Payload Services) flights, with up to 30 robotic lunar landings targeted for 2027 to deliver rovers, hoppers, and drones.

We've all heard the advice: eat your fruit and vegetables, get your vitamins, and stay healthy. For the most part, that guidance holds up. **But some nutrients have a more complicated story, and vitamin B12 is a fascinating example**, according to an article from Science Daily. Also known as cobalamin, B12 is essential for life. It helps the body produce red blood cells, keeps the nervous system functioning, and plays a central role in how cells copy and repair DNA. A 2025 case-control study from Vietnam found what researchers described as a U-shaped relationship between B12 intake and cancer risk, with both lower and higher intakes associated with increased risk. Because this kind of study can show an association but cannot prove cause and effect, the takeaway is not that B12 is dangerous. It is that balance matters. The broader message is simple: more is not always better. Cancer cannot be prevented by loading up on any single vitamin. Long-term habits matter more: eating a balanced diet, exercising regularly, avoiding smoking, protecting your skin and attending routine health screenings. So what about vitamin B12? Get enough through food or supplementation if you need it, especially if you are vegan, older or have a condition that affects absorption. But leave the megadoses on the shelf unless a doctor advises them. With B12, as with many nutrients, the goal is not as much as possible. It is the right amount.

Over the past decade, researchers studying language, cognition, and verbal processing have identified several behaviors that correlate with higher cognitive ability but that are routinely mistaken for their opposite, according to an article from Psychology Today. Two of the most compelling involve habits that many intelligent people might have been quietly apologizing for their whole lives. Although neither will win you friends at a formal dinner, both are supported by a growing and credible body of peer-reviewed research. There is a long-standing cultural assumption that people who talk to themselves are, at best, eccentric and, at worst, showing signs of something more concerning. It's the kind of behavior that invites sideways glances in supermarket aisles and prompts well-meaning family members to ask if you are OK. The research, however, tells a different story. The assumption with swearing is so embedded that it has become a kind of folk wisdom: People who swear frequently do so because they lack the vocabulary to express themselves any other way. It is, in this view, portrayed as a sign of laziness: a linguistic shortcut taken by people who cannot be bothered to find the right word. But this is almost precisely backwards.



Tom Liggett/HELIOS II

A photography student sent a 5×4 color negative into space on April 19 and exposed it to cosmic radiation, capturing a beautiful, abstract portrait of space unlike anything done before, according to an article from PetaPixel. Tom Liggett is a third-year BA (Hons) photography student at the Arts University Bournemouth (AUB) in the U.K. His groundbreaking project, HELIOS, saw him travel to New York state where he sent a series of weather balloons with negatives attached to them to altitudes of over 121,000 feet — about three times higher than commercial aircraft and

far above the protective layers of Earth's atmosphere.

Have you ever pressed a crosswalk button and wondered if it actually does anything? You might be onto something. Called placebo buttons, **controls that don't do anything exist everywhere**, according to an article from Popular Science. Sometimes it's because of accidents of history; sometimes they're installed specifically to trick people into feeling an illusion of control. Either way, they're hard to notice. Here are a few buttons you press every day that might not actually work. They are crosswalk buttons, the "close door" button on elevators, and that thermostat at your office.

The Silent Ransom Group (SRG), also known as Luna Moth, Chatty Spider, and UNC3753, is targeting law firms using social engineering techniques, according to a bulletin released by the U.S. DHS/CISA/FBI (PDF). Through phone calls and phishing emails, SRG actors pose as IT support to establish access to victim computers and exfiltrate data, usually through legitimate remote access tools or by sending an individual in-person to the victim company's location to gain physical access to computers. While SRG has victimized companies in many sectors including those in the insurance, finance, and healthcare industries, the group has consistently targeted US-based law firms since Spring 2023.

2FA isn't foolproof: Hackers still have tools to bypass your security measures and worm their way into your online spaces, through zero fault of your own. Luckily, **Google is now rolling out a new security measure that should reduce these vulnerabilities**, according to an article from Lifehacker. As long as you're running the latest version of Chrome, people looking to break into your accounts should now face a steeper uphill battle. Google officially rolled out "Device Bound Session Credentials" (DBSC) for Chrome. DBSC works by ensuring that your session cookies are stored somewhere challenging for hackers to access. Going forward, all session cookies generated in Chrome (and on other Chromium-based browsers) will be stored to your PC's Trusted Platform Module, or your Mac's Secure Enclave. These chips are designed to hold sensitive data and protect it with encryption. Only the security chip has the keys to decrypt the information there. That means even if hackers successfully infect your Mac or PC with malware, they'll have an exceedingly difficult time breaking into the security chip and stealing your session cookies.

Proton announced a new integration with Gmail that lets users send and receive Gmail messages while retaining their privacy, according to an article from Lifehacker. When users connect their Gmail to Proton Mail, Proton strips away ads, spam, and trackers from messages. The move also blocks Google from spying on how users engage with their Gmail messages. If two Gmail users are communicating via Proton Mail, their messages will be end-to-end encrypted.

A system intrusion at 7-Eleven has escalated into a major data breach, according to an article from TechRepublic. The retailer says an unauthorized third party gained access to its internal systems on April 8, exposing personal information associated with franchise applications. In breach notification letters dated May 1, the company said the attackers accessed "certain 7-Eleven systems used to store franchisee documents." 7-Eleven added that the affected files contained personal details submitted during the franchise application process, such as names, addresses, and other identifying information.

Some of you may know there's a version of UNIX for the Commodore Amiga, aptly called Amiga Unix or AMIX, starts an article from HackADay. There is an almost complete record of versions from 1.0 to 2.03, but 2.02 was lost media – until [Forgotten Computer] found it on an old Amiga. It starts with an auction held for the 40 year anniversary of the Free Software Foundation where, by just one second, the highest bidder was too late. What do you do first with an artifact as valuable as an old FSF computer? You image the hard drive. Then you make several copies, including on different computers–after all, you wouldn't want to lose the data on it. Preservation secured, the natural next thing is to boot it–and that's when we see the magic 2.02c version number. According to thorough digging by [Forgotten Computer], this version was – until now – lost.

You've used Google Search the same way for 25 years. Type something, get a list of links, click one. That era just ended. **At Google I/O last month, the company announced it's replacing its traditional search box with an AI-powered conversational engine. Instead of a list of links, Google now serves up AI-generated answers first, with built-in follow-up questions**, according to an article from eWeek. It's calling it "the biggest upgrade to Search in over 25 years." Not everyone is thrilled. DuckDuckGo saw US app installs jump an average of 18% week-over-week after Google's announcement, peaking at 30% growth on Memorial Day. iPhone installs were even wilder, averaging 33% growth with a single-day peak of nearly 70%. Traffic to DuckDuckGo's AI-free search page (noai.duckduckgo.com) grew 22.7% in the same window. DuckDuckGo CEO Gabriel Weinberg put it plainly: "Google is force-feeding AI with no way to opt out. Their results are getting worse, not better."

Brave just crossed 117.56 million monthly active users in May 2026 and DuckDuckGo says its US installs jumped 76% after Google's big AI search overhaul, according to an article from PuinikaWeb. Both privacy-focused products are basically saying the same thing in different ways. Users are moving when they feel Google is pushing too much AI into their everyday browsing and search. Brendan Eich shared fresh Brave numbers on X. Brave hit 117.56 million monthly active users in May and daily active users grew 3.1%. He called it the second-best browser growth month of 2026 and said new browser user volume hit an all-time high. This is quite a bump for a browser that already cleared 100 million MAU a little while back.

Image by Arek Socha from Pixabay

Google is rolling out a new Android security feature designed to help users spot AI-powered impersonation calls, according to an article from eWeek. The feature, called fake call detection, aims to identify situations where scammers spoof the phone number of someone in a user's contacts and use AI-generated voices to impersonate a family member, friend, employer, or other trusted person. According to Google, the feature is intended to address a growing problem as criminals increasingly combine caller ID spoofing with realistic voice-cloning technology. "Fake call detection helps protect you, your family and friends by identifying when a caller isn't who they claim to be, giving you an extra layer of defense against sophisticated AI-voice cloning scams, also called deepfake attacks, of your contacts,"

Google researchers wrote. Google describes fake call detection as a digital verification process that runs automatically in the background. When two people are using Phone by Google, the caller's device sends a silent verification signal to confirm that the call is genuinely originating from that device. The process relies on end-to-end encrypted Rich Communication Services (RCS) technology. "If a scammer tries to impersonate your contact, that initial confirmation signal will be missing," Google explained. "Your device will instantly notice this and ping your contact's actual device to double-check. If their real device says, 'I'm not making a call right now,' you'll get a warning on your screen advising you to hang up immediately."

The costs of meeting a federal mandate to make research papers freely and immediately available to the public are exorbitant, and most agencies don't have adequate plans in place to cover it, a report from the U.S. Government Accountability Office (GAO) found, according to an article from MedPageToday. The U.S. government is a huge funder of scientific research globally. In 2022, the Office of Science and Technology Policy (OSTP) issued a federal mandate to make research freely accessible to the public as soon as it's published. For this report, the GAO examined agencies' efforts to implement that mandate. Seven of nine federal agencies that the GAO reviewed issued updated plans or policies on how to meet the public access mandate, and five of those agencies' plans fully met the OSTP's guidance. The Department of Transportation and the Nuclear Regulatory Commission were still working on their plans. Making research publicly accessible comes with costs, and the GAO noted that publishers are changing their business models to adapt to a loss of subscription revenue, including by requiring authors to pay open-access fees.

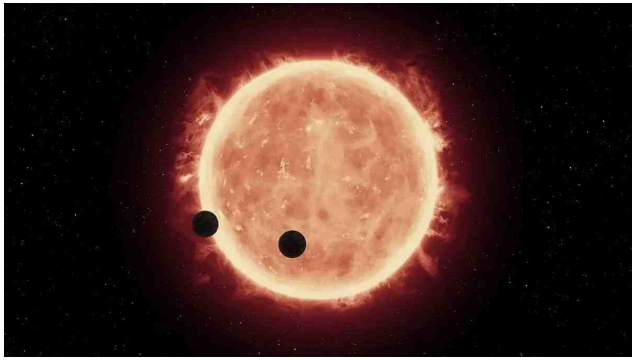
If you have (or had) a device equipped with Google Assistant, you may be eligible for some cash as part of a recent $68 million settlement, according to an article from Lifehacker. However, settlement notices are seemingly being filtered into spam folders, so users who are eligible to submit claims are likely to miss them. It isn't clear at this time how much individual claimants will receive, but the settlement does use a "points" system depending on your situation. Claims must be filed by Aug. 27, and a final approval hearing is scheduled for Oct. 1.

According to multiple polling services, approximately 70% of Americans are opposed to having AI data centers in their communities. From a news release by Gallup, seven in 10 Americans oppose constructing data centers for artificial intelligence in their local area, including nearly half, 48%, who are strongly opposed. Barely a quarter favor these projects, with 7% strongly in favor. A study done by Embold Research reveals that opposition to local data center construction has risen significantly just over the past six months. In December 2025, just over half of voters (52%) opposed building a data center in their

local area, with 36% supportive. By May 2026, total opposition had climbed to 70%, with support falling to just 21%. Perhaps most striking, the share of voters who strongly oppose local construction grew by more than 20 percentage points, rising from 36% to 58%.

Research suggests that equal servings of animal protein may pack a much bigger muscle-building punch than their plant-based counterparts, according to an article from SciTechDaily. When it comes to protein, the same serving size on paper may not mean the same nutritional payoff in the body. A 2023 Purdue University study found that two ounce equivalents (oz-eq) of animal-based protein foods supplied more bioavailable essential amino acids (EAA) than the same two oz-eq amount of plant-based protein foods. Essential amino acids are especially important because the body cannot make them on its own. They must come from food, and they help support muscle and whole-body protein building. The findings add a sharper edge to a familiar nutrition question: are all protein foods truly comparable when they are measured by the same serving system?

The use of dietary supplements has increased sharply in recent years. **Vitamins, minerals and other nutritional products are often marketed as simple ways to boost energy, support immunity, protect brain health or even promote longevity**, according to an article from ScienceDaily. For many people, taking supplements can feel like a sensible, proactive health habit. But this perception can be misleading. For people who already have adequate nutrition, many supplements offer little or no measurable benefit. Some are simply an unnecessary expense. Others are not risk-free: high doses of certain vitamins and minerals can cause toxicity, interfere with medications or produce unintended health effects. For older adults, however, the picture is more complicated. The most useful question is not simply whether supplements are "good" or "bad", but whether someone is actually deficient, what might be causing that deficiency and whether a supplement is the safest way to address it.



(*NASA/ESA/STScI/J. de Wit/MIT*)

If someone turns up with crumbs on their chin, it's natural to wonder where the cookies went. Astronomers have found themselves asking that same question about a handful of very strange stars. **Among thousands of stars studied by astronomers, six red dwarfs stood out for carrying traces of a strange element in their atmospheres**, according to an article from ScienceAlert. In normal circumstances, this element should long ago have been annihilated deep within the stars' interiors. Its presence here suggests that these six stars have been raiding the cookie jar – if the cookie jar were full of Earth-like planets.

We usually think of our Solar System as a calm, well-organized family of planets, but a new study suggests its early days may have been far more chaotic. **Instead of the four giant planets we know today, scientists think there may have once been six, meaning two massive worlds could have been kicked out of the Solar System entirely, drifting off into deep space**, according to an article from GeekSpin. Even more surprising, researchers say clues to what happened may be hidden in the moons of Uranus, hinting at a story that doesn't quite fit existing theories. The findings are challenging what we know about how our cosmic neighborhood formed—and raising new questions about where those lost planets might have gone.

New evidence suggests calcium and vitamin D supplements may do far less to prevent fractures and falls than widely believed, according to an article from SciTechDaily. Calcium and vitamin D supplements, whether taken separately or together, provide little to no meaningful benefit in preventing fractures or falls in most older adults, according to a major review published in *The BMJ*. Nearly one in three adults age 65 and older experiences a fall each year. Many of these falls lead to fractures, which can cause pain, lower quality of life, and increase the need for assisted living or residential care. As a result, reducing falls and fractures remains a major public health goal worldwide. Earlier reviews have also found little evidence that calcium or vitamin D supplements

reduce fracture risk, and findings on combined supplementation have been inconsistent. The role of vitamin D in preventing falls has also remained uncertain. Even so, many doctors, health guidelines, and regulatory agencies continue to recommend vitamin D supplements, with or without calcium, to support bone health. Prescriptions for these supplements have also risen significantly in recent years.

So, how do you scan a QR code when it's already on your phone's screen? That's the question that an article from BGR answers. It's surprisingly easy using iPhone and Android devices. The first step is to take a screenshot of the QR code and then open the image in your gallery. On a typical Android, to take a screenshot, press and hold the power and volume down button simultaneously. The screen will flash white to indicate the capture. On most iPhones, press and hold the wake and volume up button until the screen flashes. Then, open the screenshot of the code in your phone's gallery. To see the code's message on an iPhone, tap and hold on the QR code to open a menu with viewing options. With Android, tap the image of the QR code and then the Lens icon. In either case, you should see the QR code's message or the URL it's directing you to visit.

A surprising gut-brain discovery suggests that anxiety could one day be treated with specially designed probiotics, according to an article from SciTechDaily. Could anxiety be shaped, at least in part, by tiny organisms living in the gut? Research from Duke-NUS Medical School and the National Neuroscience Institute of Singapore points to a striking connection between gut microbes and anxiety-related behavior. The findings suggest that certain compounds made by gut bacteria, especially molecules called indoles, can influence brain activity involved in fear, stress, and emotional balance. The study, published in EMBO Molecular Medicine, adds to a growing body of research showing that mental health is not controlled by the brain alone. Instead, the gut and brain appear to be in constant communication, with microbes helping to shape some of the chemical signals that affect mood and stress responses.

A vitamin D-based therapy may help remodel the protective barrier surrounding pancreatic tumors, offering a potential new treatment strategy, according to an article from SciTechDaily. Pancreatic cancer is notoriously difficult to treat in part because tumors surround themselves with a dense, protective barrier that blocks drugs and suppresses immune activity. Now, a small clinical trial led by researchers at Dana-Farber Cancer Institute suggests that an FDA-approved vitamin D analog may help dismantle some of those defenses, potentially making tumors more vulnerable to treatment. The study, published in Nature Cancer, enrolled patients with previously untreated metastatic pancreatic cancer who received standard chemotherapy either with or without paricalcitol, a vitamin D analog already approved by the FDA for other conditions. Researchers found that adding paricalcitol, given either orally or intravenously, was safe and reduced fibroblast activity within the tumor microenvironment, confirming earlier findings from Salk laboratory studies.

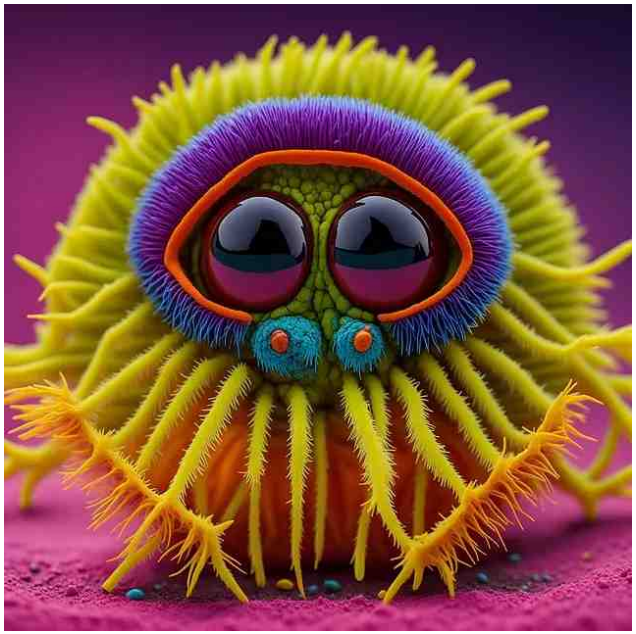


Image by Pete Linforth from Pixabay

A use-after-free vulnerability in the Linux kernel's nftables subsystem has been

disclosed, enabling unprivileged local attackers to escalate privileges to root on widely deployed distributions including Debian Bookworm, Debian Trixie, Ubuntu 22.04 LTS, and Ubuntu 24.04 LTS, according to an article from CyberSecurityNews. Tracked as CVE-2026-23111, the flaw was discovered in early 2025 and patched upstream on February 5, 2026, via a kernel commit. Security researcher Oliver Sieber of Exodus Intelligence published a detailed technical write-up alongside a working exploit demonstrating >99% reliability on idle systems. The bug originates in the `nft_map_catchall_activate()` function within the nftables subsystem — a packet filtering framework built on top of Linux's Netfilter hooks. Specifically, a single inverted conditional check (a misplaced ! operator) causes the function to incorrectly skip inactive catchall elements during the abort process, instead of reactivating them.

Another month, another set of vulnerabilities for Google Chrome. Whodathunkit? According to a report by Malwarebytes Labs, Google's latest update for Chrome (version 149.0.7827.102/.103 for Windows and Mac and 149.0.7827.102 for Linux) includes patches for 74 security vulnerabilities, says an article from Lifehacker. 17 of these vulnerabilities are rated as "Critical," while all but two of the others are rated "High." That alone would indicate this update is rather important, but there's more: One of these vulnerabilities is currently actively exploited. Google confirmed in its update that there is an exploit in the wild for the flaw tracked as CVE-2026-11645. This flaw is an "out of bounds memory access" vulnerability affecting V8, Chrome's JavaScript engine. Hackers can exploit the flaw to have their own program read or write data outside of the memory spaces it's supposed to access. In other words, a hacker could run their own code in Chrome, as if it were something legitimate Google placed in the browser. All a hacker would need to do is trick you into clicking a link to a malicious website, and they could effectively take over your browser. Because this flaw was exploited before Google issued the patch, it's considered a "zero-day." That's dangerous, because hackers could potentially abuse the flaw en masse before most users have a chance to update their browsers. If there is a silver lining here—besides there being an update available to patch the flaw—it's that the zero-day is limited to Chrome

A dedicated cross-platform password manager like Bitwarden can really help, according to an article from Lifehacker. It's open-source, encrypted by default, and most of its features are free on all platforms. And you can even use it to sync passkeys and two-factor authentication codes between all your platforms. But just using the Bitwarden apps to store passwords and autofill them won't get you very far. Because you're dealing with extremely sensitive passwords and secure notes, you should take some extra steps to secure your data, and make the password management process a bit easier (without compromising your security). Bitwarden is hiding powerful features that work across all platforms.

PCLinuxOS Recipe Corner

Chicken Lazone

Serves: 4

INGREDIENTS:

1 1/2 teaspoons chili powder
1 1/2 teaspoons smoked paprika
1/4 teaspoon cayenne pepper
1/2 teaspoon freshly ground black pepper
2 teaspoons garlic powder
1 1/2 teaspoons onion powder
4 skinless, boneless chicken breasts
1 1/2 teaspoons kosher salt, or to taste
1 tablespoon vegetable oil
1/2 lemon, juiced
2/3 cup heavy cream
1/3 cup water
1/4 cup sliced green onions
1 tablespoon cold butter, cubed

DIRECTIONS:

Mix chili powder, smoked paprika, cayenne, black pepper, garlic powder, and onion powder together in a small bowl; set aside. Half will be used for the chicken rub and half for the sauce.

Season chicken on both sides with salt. Sprinkle half the spice mixture over both sides of the chicken. Wrap and refrigerate for 1 hour (or up to overnight), or until ready to cook.

Heat oil in a heavy-duty nonstick pan over medium-high heat. Sear chicken, turning once, about 3 to 4 minutes per side; chicken will be about 75% cooked at this point, and will finish cooking in the sauce. Turn off heat; transfer chicken to a plate and set aside.

Add remaining spice mixture to the pan and stir around in the hot oil for 30 seconds. Add lemon juice, cream, and water. Stir in a pinch of salt.

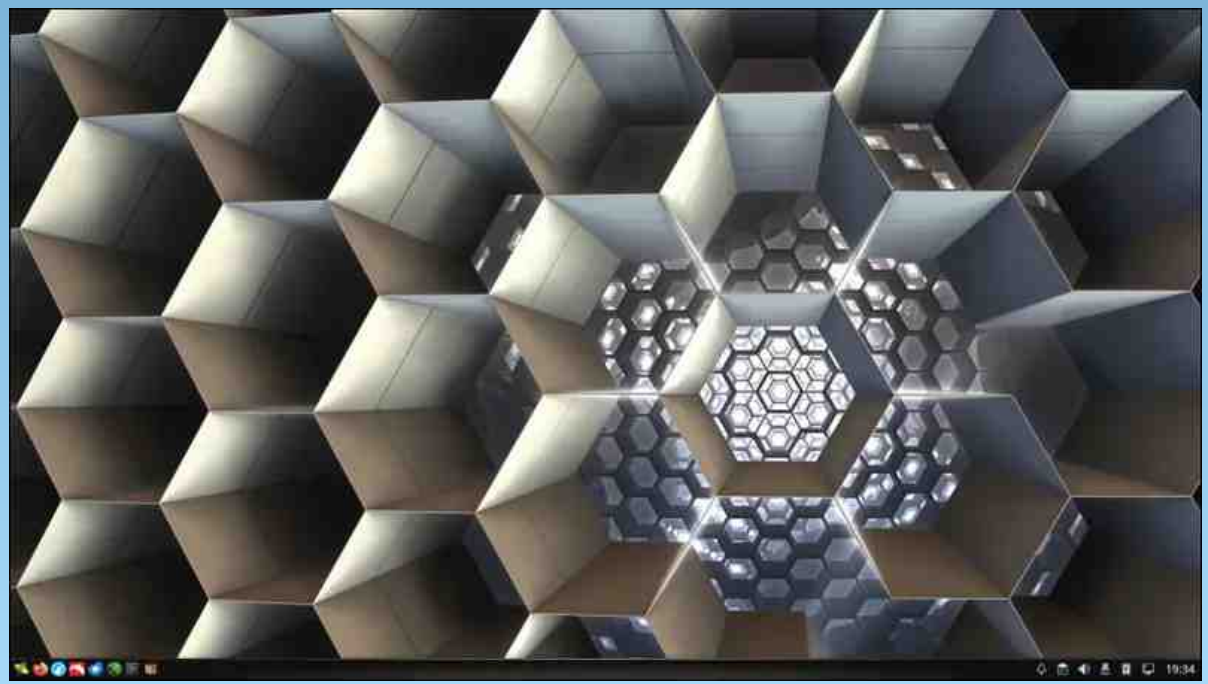
Bring to a simmer over medium-high heat, stirring occasionally. Reduce to medium-low and return chicken to the pan along with any accumulated juices. Simmer gently, turning occasionally to coat with sauce, until chicken is no longer pink at the center and juices run clear, 8 to 10 minutes. Time will vary based on the size of chicken breasts. An instant-read thermometer inserted near the center will read 165 degrees F (74 degrees C).

Reduce heat to low and add green onions and butter. Stir into the sauce, while also basting chicken with the sauce. Taste for seasoning and serve.

NUTRITION:

Calories: 411 Carbs: 6g Fiber: 2g
Sodium: 627mg Protein: 39G

Screenshot Showcase



Posted by luikki, on June 2, 2026, running KDE.

Alternative "Fixes" For The NLUUG Repo Path Problem

by David Pardue (kalwisti)

Last month we published an article about what happened during the Great NLUUG Snafu (GNS) of mid-May 2026, with instructions on how to manually fix the repo directory path so that your PCLinuxOS computer can receive updates. For more detailed background on the GNS, please consult the June 2026 article.

If you have not yet made the manual change to your system configuration, I will present additional options for you to consider. This proves the truism that there are multiple ways to accomplish a task on Linux.

Texstar has released new, fully updated ISOs — KDE 2026.05.27, KDE Darkstar 2026.05.31, MATE 2026.05.29 and Xfce 2026.05.30 — which fix the software repository links and removed conflicting libraries (i.e., the "lib64pangomm" issue). If you want to start fresh with a new installation, this is the most expeditious option. (You should safely back up your personal files before proceeding, of course.)

If you prefer a manual fix, there are two variant methods to accomplish this task. I have tested each method several times in VirtualBox and they both work as expected. I can (hopefully) encourage you by stating that both procedures are straightforward. It takes longer to describe and read through them, than it does to actually apply the fixes.

Check If DNF Is Installed

Both procedures described here assume that you have dnf and/or the DNF Package Manager GUI installed on your system. To check whether these tools are installed, issue the command below from a Terminal/Konsole:

```
$ dnf list --installed | grep dnf
```

If dnf is installed, you will see something like this:

```
[david@pclos-openbox ~]$ dnf list --installed | grep
dnf
 dnf.x86_64 5.4.2.1-1pclos2026 x86_64
 dnf-package-manager.x86_64 1.14-7pclos2026 x86_64
 dnfzypp-sources-list.x86_64 1.0-5pclos2025 <unknown>
 lib64dnf5.x86_64 5.4.2.1-1pclos2026 x86_64
 lib64dnf5-cli.x86_64 5.4.2.1-1pclos2026 x86_64
```

If dnf is not installed, you will see the output below:

```
[david@pclos-mate-old-test ~]$ dnf list --installed |
grep dnf
 bash: dnf: command not found
```

If your system is outdated and only has apt/Synaptic installed, please refer to the instructions in my June 2026 article. Remember that **Synaptic is no longer supported** in PCLinuxOS; therefore, updating via Synaptic is trickier and you may encounter issues. In this situation, Synaptic should be used as a temporary bridge to update your system — and to install dnf. Afterwards, you should immediately switch to DNF. If you run into trouble, it may be easier to perform a fresh installation (*personal opinion*).

I began experimenting with these variant approaches because (a). I noticed that the PCLinuxOS KnowledgeBase article was updated to include another method to fix the repo path ["**Method A**" below]; and (b). Forum member margarita hinted at a solution using a similar method ["**Method B**" below].

Each method requires light use of the Terminal/Konsole. They are about equal in terms of difficulty (in my estimation). If you carefully follow the instructions, you will succeed at repairing your system.

Tip: I recommend reading through the entire procedure before deciding which method to use. If you are a visual problem-solver, there is a flowchart at the end of the article which outlines the major steps of each procedure.

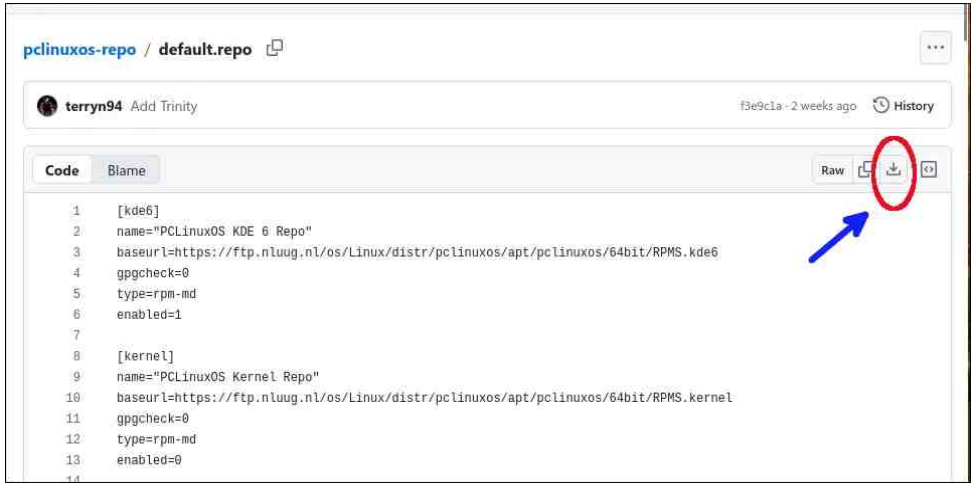
Notice that the two methods are identical at their core: a "default.repo" file which contains a correct, updated directory path for the NLUUG repository. Only the delivery/installation method differs between the two approaches.

1.a. Method A (PCLinuxOS Wiki)

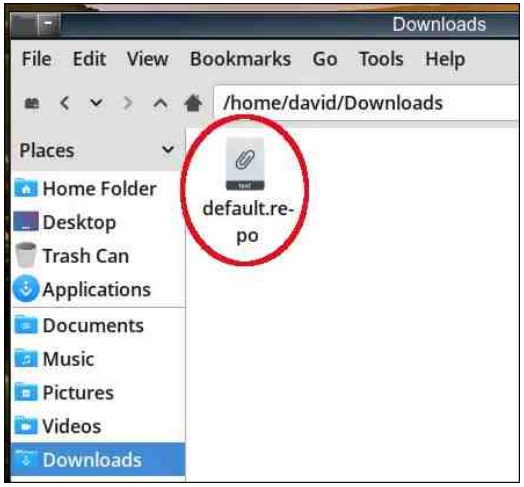
Initial Steps

Download the correct repository configuration file from terryn's GitHub.

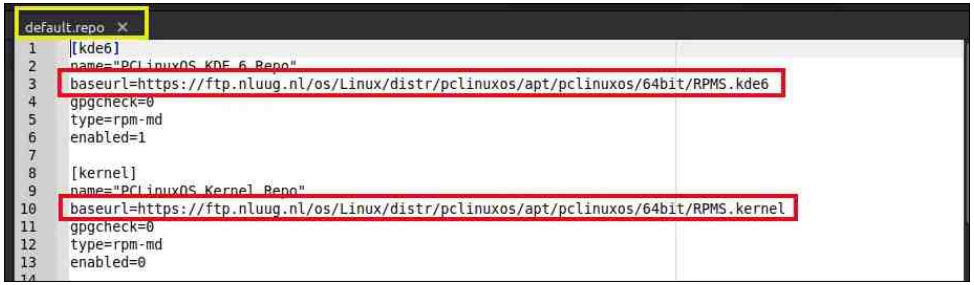
Click on the Download icon (in the upper right corner).



The file will be saved to your Downloads folder (right, top).



Note: Opening the "default.repo" file with a text editor will show that it contains the correct, updated directory path for the NLUUG repository.



Open a Terminal and type the commands below (separately):

```
cd Downloads
su -c "cp default.repo /etc/repo.d"
```

Enter your root password if/when prompted.

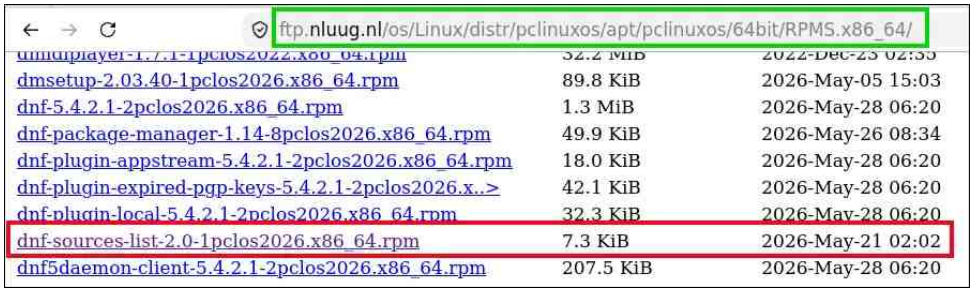
(The command above will replace your current " /etc/repo.d/default.repo " with the file you just downloaded from GitHub.)

Next, jump to Section 2, "Subsequent Steps: Update via DNF PM."

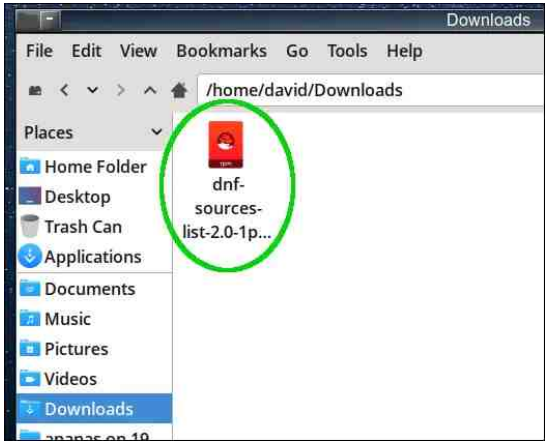
1.b. Method B ("margarita Method")

Initial Steps

Download the "dnf-sources-list-2.0-1" .rpm package from the link below:
https://ftp.nluug.nl/os/Linux/distr/pclinuxos/apt/pclinuxos/64bit/RPMS.x86_64/dnf-sources-list-2.0-1pclos2026.x86_64.rpm



This file will be saved to your Downloads folder.



Note: You do **not** need to extract the contents of this .rpm package. However, for the purpose of demonstration, I would like to show that it contains a "default.repo" file with the correct, updated directory path for the NLUUG repository (right, top).

Next, you will install the new “dnf-sources-list” package with this command sequence. Open a Terminal/Konsole and type:

(To navigate to your Downloads folder)

```
cd Downloads
```

(To install that .rpm file, type the following command.)

```
su -c "rpm -Uvh dnf-sources-list*.rpm"
```

Enter the root password when prompted.

You should see output similar to this:

```
[david@pclos-kde-test-vb Downloads]$ su -c "rpm -Uvh
dnf-sources-list*.rpm"
Password:
Verifying...
########################################### [100%]
Preparing...
########################################### [100%]
Updating / installing...
1: dnf-sources-list-2.0-1pclos2026
########################################### [ 50%]
Cleaning up / removing...
```

```
2: dnfzypp-sources-list-1.0-5pclos2025
########################################## [100%]
[david@pclos-kde-test-vb Downloads]$
```

To close the Terminal/Konsole, type "**exit**" when the command has run.

Next, jump to Section 2, "Subsequent Steps: Update via DNF PM."

2. Subsequent Steps: Update via DNF PM

Open the DNF Package Manager GUI.

Click on the **Refresh** button and wait a few moments.

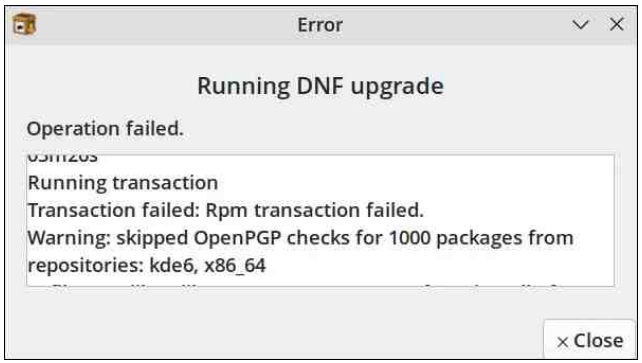
Click on the **Select All** button. Then click on the **Update** button to begin the update process.

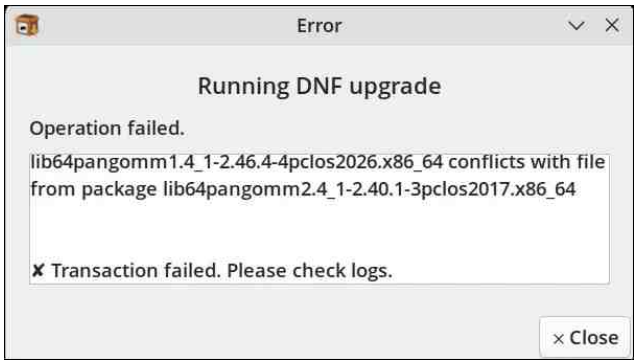
During the update process, you will probably encounter the so-called "lib64pangomm conflict" that will cause the DNF transaction to fail. Don't worry! There is a fix for it, too.

If the update fails, jump to Section 3, "lib64pangomm Error?"

3. lib64pangomm Error?

If the update fails, you will be notified via an Error message.





Remain calm because Upgreyed has kindly provided a fix for this situation (below).

Close the DNF PM GUI.

Open a Terminal/Konsole and acquire root privileges by typing "su -" [omit the quotation marks; there is a single space and a single hyphen after su].

Enter the root password.

Type the command below:

```
rpm -e lib64pangomm2.4_1-2.40.1-3pclos2017 --nodeps
```

Tip: In the command above, there should be two hyphens preceding the "nodeps" option. When the command runs successfully, you will see no output. No output is normal/expected in this case.

You should see something like the output below:

```
[david@pclos-kde-test-vb ~]$ su -
Password:
[root@pclos-kde-test-vb ~]# rpm -e lib64pangomm2.4_1-
2.40.1-3pclos2017 --nodeps
[root@pclos-kde-test-vb ~]#
```

Type "**`exit`**" (without quotation marks) twice to exit and close the Terminal/Konsole.

Open the DNF PM GUI again. Press the **Refresh** button.

Click on the **Select All** button, then click on the **Update** button.

The packages which you previously downloaded are already cached on your system, so you do not have to download them all over again.

DNF should begin installing and upgrading packages. Wait patiently for the upgrade to finish.

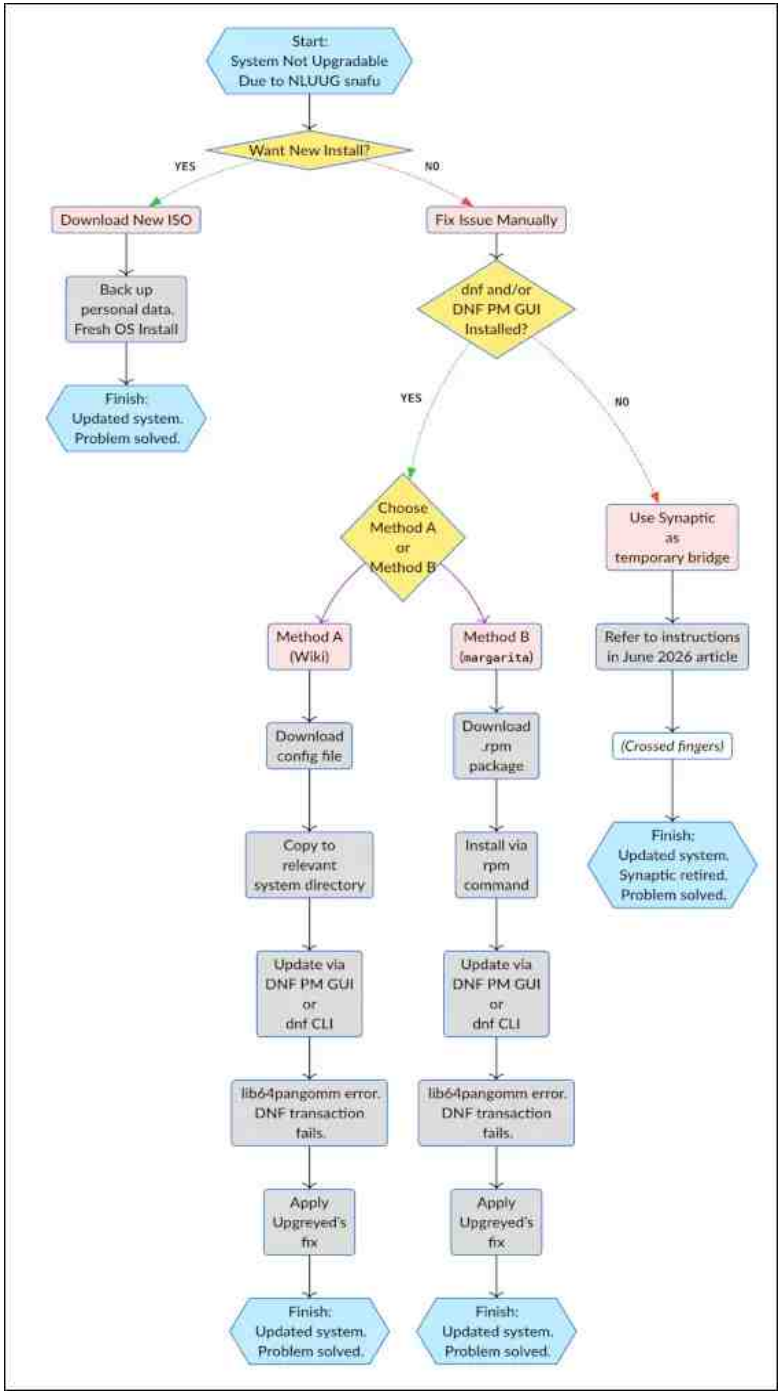
Close/shut down the DNF PM GUI.

Log out of your current session and reboot your computer.

4. Switch to a Different Repo Mirror (*Optional*)

At this point, everything should be back to normal. You may change your repo mirror from NLUUG to one that is faster and/or geographically closer to you (if you wish). You can install two small GUI utilities (created by Upgreyed) to help with these tasks: "fastrepo" and "my-repo-changer". They both work well.

I hope the flowchart at right will help you visualize the major steps in each procedure, and help you choose which method to use. I created the flowchart with Typst, using the fletcher package which is designed to draw diagrams with nodes and arrows. (You can download a full-sized version of the flowchart from my Box.com account, if you are interested.



Disclaimer

1. All the contents of the PCLinuxOS Magazine are only for general information and/or use. Such contents do not constitute advice and should not be relied upon in making (or refraining from making) any decision. Any specific advice or replies to queries in any part of the magazine is/are the person opinion of such experts/consultants/persons and are not subscribed to by the PCLinuxOS Magazine.

2. The information in the PCLinuxOS Magazine is provided on an "AS IS" basis, and all warranties, expressed or implied of any kind, regarding any matter pertaining to any information, advice or replies are disclaimed and excluded.

3. The PCLinuxOS Magazine and its associates shall not be liable, at any time, for damages (including, but not limited to, without limitation, damages of any kind) arising in contract, rot or otherwise, from the use of or inability to use the magazine, or any of its contents, or from any action taken (or refrained from being taken) as a result of using the magazine or any such contents or for any failure of performance, error, omission, interruption, deletion, defect, delay in operation or transmission, computer virus, communications line failure, theft or destruction or unauthorized access to, alteration of, or use of information contained on the magazine.

4. No representations, warranties or guarantees whatsoever are made as to the accuracy, adequacy, reliability, completeness, suitability, or applicability of the information to a particular situation.

5. Certain links on the magazine lead to resources located on servers maintained by third parties over whom the PCLinuxOS Magazine has no control or connection, business or otherwise. These sites are external to the PCLinuxOS Magazine and by visiting these, you are doing so of your own accord and assume all responsibility and liability for such action.Material Submitted by UsersA majority of sections in the magazine contain materials submitted by users. The PCLinuxOS Magazine accepts no responsibility for the content, accuracy, conformity to applicable laws of such material.

Entire Agreement: These terms constitute the entire agreement between the parties with respect to the subject matter hereof and supersedes and replaces all prior or contemporaneous understandings or agreements, written or oral, regarding such subject matter.

Screenshot Showcase

Posted by Meemaw, on June 4, 2026, running Xfce.

A Custom Script To Manage Your Wireguard VPN Connection

by Cuig

Why write this script at all?

Because I like simple, non-intrusive ways of getting things done and, for various reasons, other options open to me were not acceptable.

Main amongst those is that Proton, my selected VPN provider, only provides their application for distros that have systemd running. There is a very short list of distros they "support".

An alternative called wireguird (note spelling), a nice GUI to manage wireguard VPNs, has had a problem on my installs, in that when a VPN is deactivated it leaves me without an internet connection!

So that is my reason for this script. But why a VPN at all?

Why do I want to use a VPN? Several reasons really, most of which have been covered in other pages of this magazine, but mostly because of my government's actions.

- Lots of foreign sites (example: rt.com) are blocked in my country in an attempt to allow me to read ONLY what my government approves.

- That is not acceptable to me under any circumstances.

- That and the added privacy of using a VPN is why I use one.

- I might occasionally view public service (free-to-air) TV services from other locales, which require use of a VPN.

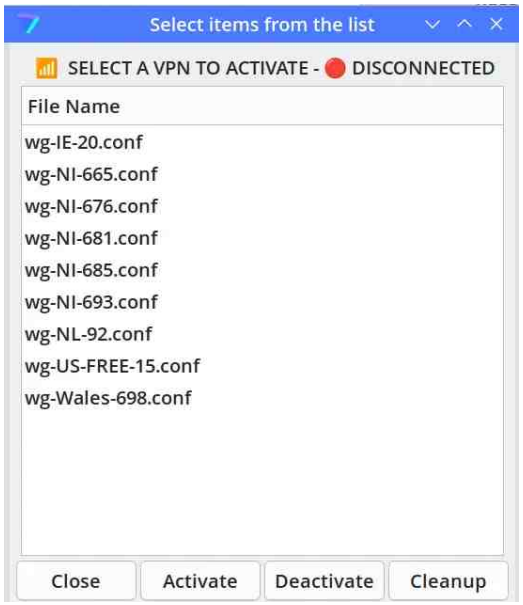
I use the wireguard protocol as it is the most recent, secure and fastest of those generally available. I use SUDO in this script as the wireguard.conf files are stored in /etc/wireguard and are owned by root and some of the commands used need root privileges. If the permissions of these wireguard files are wrong, an error is thrown - presumably because this would reduce the security.

I have used Zenity for the GUI as there was something I wanted to do that I could not figure out in Yad (which I tried first).

Be aware that SUDO needs to be set up for the various elements used in this script. Besides doing that, the dependencies are Zenity, SUDO, wireguard-tools, and in my case for the icon, papirus-icon-theme.

The Script

The script will select your primary network connection and only permit one VPN connection to be active at any time.

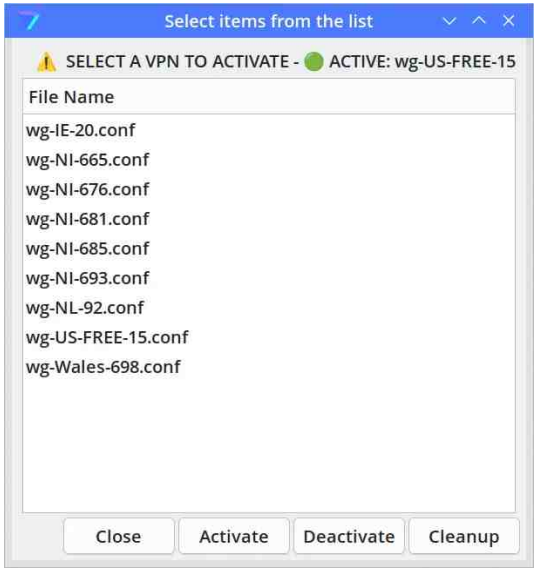


On launch with no VPN active, this is shown. It lists the files with ".conf" extension in the /etc/wireguard directory. These are the files one gets from the VPN service provider. Yours of course will be different to mine shown below.

As can be seen from the "Disconnected" at the top, there is no VPN active and you are invited to select one to activate. Having made your selection, simply click the "Activate" button and the VPN is activated.

A window is shown to confirm which VPN is activated. Select OK to close this.

On the next launch of the script the active VPN will be shown under the title.



From this window, you can either Deactivate the VPN, returning you to your normal IP address, or you can select another VPN file and click Activate.

Selecting a new file and the “Activate” button will Deactivate the existing VPN and Activate the newly chosen one, ensuring you have only one VPN active at any time. A small window will inform you of the name of the new VPN you chose.

Selecting the Deactivate button will bring up the following window.

On the main window, the "Close" button is obvious, but the "Cleanup" button deserves some explanation. In case of some mishap, such as a PC freeze or hard shutdown, using the "Cleanup" button should get things back to normal. This option was added to help resolve some fault or failure. It essentially resets your connection back to default. It could happen that your desktop freezes due to some other application or process going wrong and your VPN network is left in a confused state. This should help if this should occur.

I use Proton VPN, so I used their icon for the script. The script is easily editable to change the icon – just one entry to be changed.

There is not much else to say really. This activates, switches and deactivates VPN connections using a simple interface. Nothing fancy; just functional.

I included a desktop file for convenience. I placed the working script in $HOME/bin/ProtonWG, so if you place it elsewhere, the path in the desktop file will need to be edited.

First here is the .desktop file, my-vpn-manager.desktop:

```
#!/usr/bin/env xdg-open
[Desktop Entry]
Categories=Internet;Network;
Comment[en_GB]=VPN Manager
Comment=VPN Manager
Exec=$HOME/bin/ProtonWG/my-vpn-manager
GenericName[en_GB]=VPN Manager
GenericName=VPN Manager
Icon=/usr/share/icons/Papirus/48x48/apps/proton-vpn-logo.svg
MimeType=
Name[en_GB]=my-vpn-manager
Name=my-vpn-manager
Path=
StartupNotify=true
Terminal=false
TerminalOptions=
```

```
Type=Application
X-KDE-SubstituteUID=false
X-KDE-Username=
```

Next is the working script, my-vpn-manager.

```
#!/bin/bash
# ============================================================================
# VPN Manager Script v1.6 2026-04-13
# Features: Switching VPN, Stale VPN cleanup, File detection of ".conf"
only
# OS: PCLinuxOS 2026
# Dependencies: Sudo, wireguard-tools, Zenity, papirus-icon-theme
# ============================================================================

set -x         # For debug output in a terminal
cd /etc/wireguard
Icon=/usr/share/icons/Papirus/48x48/apps/proton-vpn-logo.svg

# --- Helper: Detect Active VPN ---
get_active_vpn() {
    # Must use sudo to read kernel state. We need the 2nd field.
    local iface=$(sudo wg show 2>/dev/null | awk 'NR==1 {print $2}')
    echo "$iface"
}

# --- Helper: Find Primary Network Interface ---
get_primary_iface() {
    # Try to find the interface with the default route
    local iface=$(ip route | grep default | awk '{print $5}' | head -n1)

    # Fallback: First non-loopback interface
    if [[ -z "$iface" ]]; then
            iface=$(ip link show 2>/dev/null | awk -F': ' '/^[0-9]+: (eth|
wl|en)/ {print $2}' | head -n1)
    fi
# Last resort: eth0
    if [[ -z "$iface" ]]; then
        iface="eth0"
    fi
    echo "$iface"
}

# --- Helper: Cleanup ALL Stale Connections ---
do_cleanup() {
    echo ">>> Running Cleanup..."

# Extract ONLY the interface names (first column of first line for each
block)
# Use 'awk' to find lines starting with "interface:" and print the 2nd
field.
local ifaces=$(sudo wg show 2>/dev/null | awk '/^interface:/ {print $2}')

if [[ -z "$ifaces" ]]; then
    zenity --window-icon=$Icon --info --text="  No active connections
found."
    return
fi

local count=0
for i in $ifaces; do
    echo "Stopping: $i"
    sudo wg-quick down "$i" 2>/dev/null || true
    ((count++))
done

local PRIMARY=$(get_primary_iface)
echo "Resetting network interface: $PRIMARY"
if [ -f /usr/libexec/nm-ifdown ] && [ -f /usr/libexec/nm-ifup ]; then
    sudo /usr/libexec/nm-ifdown "$PRIMARY"; sleep 1; sudo /usr/libexec/nm-
ifup "$PRIMARY"
else
    sudo ip link set "$PRIMARY" down; sleep 1; sudo ip link set "$PRIMARY"
up
fi
```

```
    zenity --window-icon=$Icon --info --text="  Cleaned up $count
connection(s)."
    sleep 1
}

# --- Helper: Hard Kill ---
do_hard_kill() {
    local iface=$1
    sudo wg-quick down "$iface"
    local PRIMARY=$(get_primary_iface)
    if [ -f /usr/libexec/nm-ifdown ] && [ -f /usr/libexec/nm-ifup ]; then
        sudo /usr/libexec/nm-ifdown "$PRIMARY"; sleep 1; sudo /usr/libexec/
nm-ifup         "$PRIMARY"
    else
        sudo ip link set "$PRIMARY" down; sleep 1; sudo ip link set
"$PRIMARY" up
    fi
}

#
================================================================================
# MAIN LOOP
#
================================================================================
while true; do
    # 1. Check Status (Refreshed every loop)
    CURRENT=$(get_active_vpn)

if [[ -n "$CURRENT" ]]; then
    TITLE="⚠️ SELECT A VPN TO ACTIVATE - 🟢 ACTIVE: $CURRENT"
else
    TITLE="📶 SELECT A VPN TO ACTIVATE - 🔴 DISCONNECTED"
fi

    # 2. Show Menu
    SELECTED=$(ls *.conf 2>/dev/null | zenity --window-icon=$Icon --
text="$TITLE" --list --height 600 --column="File Name" --ok-
label="Activate" --extra-button="Deactivate" --extra-button="Cleanup" --
cancel-label="Close")

    EXIT_CODE=$?

    # 3. Cancel
    if [ $EXIT_CODE -eq 1 ] && [ -z "$SELECTED" ]; then
        exit 0
    fi

    # 4. Cleanup Button
    if [ "$SELECTED" = "Cleanup" ]; then
        do_cleanup
        continue
    fi

    # 5. Deactivate Button
    if [ "$SELECTED" = "Deactivate" ]; then
        if [[ -n "$CURRENT" ]]; then
            do_hard_kill "$CURRENT"
            zenity --window-icon=$Icon --info --text="✅ $CURRENT
deactivated."
        else
            zenity --window-icon=$Icon --info --text="⚠️ No VPN active."
        fi
        continue
fi

    # 6. Handle Activate Button - new or switch VPN
    if [ $EXIT_CODE -eq 0 ]; then
        if [ -z "$SELECTED" ]; then
            zenity --window-icon=$Icon --warning --text="⚠️ No file
selected."
            continue
    fi

    # If a VPN is active, stop it silently first
    if [[ -n "$CURRENT" ]]; then
        echo "Switching from $CURRENT to $SELECTED..."
```

```
        sudo wg-quick down "$CURRENT"
        # Wait for kernel to fully clear routes (Critical for stability)
        sleep 2
    fi
    break
fi
done

# --- Final Activation ---
echo "  Activating: $SELECTED"
FILE="${SELECTED%.*}"

    # Safety Check: Ensure no other VPN is active before starting
    sleep 1
    FINAL_CHECK=$(get_active_vpn)

    if [[ -n "$FINAL_CHECK" ]]; then
    # If the same interface is still there, try one last time to stop it
    if [[ "$FINAL_CHECK" == "$CURRENT" ]]; then
        echo "Warning: Interface $FINAL_CHECK still active. Retrying
stop..."
        sudo wg-quick down "$FINAL_CHECK"
        sleep 1
        FINAL_CHECK=$(get_active_vpn)
    fi

    if [[ -n "$FINAL_CHECK" ]]; then
        zenity --window-icon=$Icon --error --text="⛔ Failed: Another VPN
($FINAL_CHECK) is still active."
        exit 1
    fi
fi

sudo wg-quick up "$FILE"

if [ $? -eq 0 ]; then
    zenity --window-icon=$Icon --info --text="✅ VPN $FILE activated
successfully."
else
    zenity --window-icon=$Icon --error --text="❌ Failed to activate $FILE."
fi
```

You can download the bash script from the magazine server, here. Store the file in the directory where you normally store your bash scripts. Be sure to change the filename from wireguard-vpn-manager.sh.txt to wireguard-vpn-manager.sh, and to make the file executable. The bash script is 5.2 KiB in size, so it should download quite quickly.

A Note About SUDO

One aspect of this exercise I had forgotten about was assigning necessary sudo privileges to the user. This is done by a couple of simple edits to the **/etc/sudoers** file.

As we know, to gain full privileges we su to root, and this is achieved by this entry in the file: **root ALL=(ALL:ALL) ALL**

So if we want our user to have similar privileges we add this entry: **user ALL=(ALL:ALL) ALL**

Because we are using sudo within a script we have no means of knowing when it asks for a password, so we specify that sudo does not require a password for that user: **user ALL=(ALL:ALL) NOPASSWD: ALL**

This has the effect of sudo never requiring a password from the 'user' specified. Other users will still need to enter their password.

For more specific control over what commands to allow without password, the sudoers entry can be something like this:
user ALL=(ALL) NOPASSWD: /usr/bin/wg, /usr/bin/wg-quick, /sbin/ip, /usr/bin/nmcli, /usr/libexec/nm-ifup, /usr/libexec/nm-ifdown, /sbin/dhclient

This gives the user specified the use of those commands, using sudo, but without entering a password and so should work within the scripts. All

other commands, requiring elevated privileges, will necessitate inputting the password. The blanket entry needs to be present to allow use of SUDO for commands other than specified ..... **user ALL=(ALL:ALL) NOPASSWD: ALL**

I believe that the line above specifying the commands should be sufficient for running the posted scripts.

Further update!

After posting the original bash script (shown above), Cuig went on to recreate the script as a Python script. The Python script has the added benefit of dynamically showing, via the icon in the notification area, whether the Wireguard VPN connection is currently on or off.

You can find the Python script here in the PCLinuxOS forum. The Python script is a little bit larger (approximately 64 KiB, and the script is split between two posts due to its size). You can also download the Python version of the script from the magazine server, here. Just as with the bash scripts you download from the magazine website, store the file in the directory where you normally store your scripts, remove the ".txt" file extension, and make the file executable. To run it, the command should be "python3 wireguard-vpn-manager.py".

GIMP Tutorial: The Resynthesizer Plug-In

by Meemaw

Three years ago, I did a tutorial reviewing the Resynthesizer plug-in that GIMP has. It's an interesting plug-in which allows you to seamlessly remove something from an image that you don't want. Since GIMP was updated to version 3.0, those older plug-ins no longer work, so I went looking. I did find a tutorial from Davies Media Design. Davies does a good job of explaining what the resynthesizer does. However, this was for GIMP 2.10.

I also looked on our forum. Forum member semperOSS posted this how-to, but it didn't work for me. Then I went to the GIMP Forum and found this thread. You can download a file named resynth-linux-deb.zip, unzip it and put the 4 folders in your Gimp 3.2 plugins folder.

(For Linux users) ~/.config/GIMP/3.2/plug-ins/
(For Windows users)
Users\YourUserName\AppData\Roaming\GIMP\3.2\plug-ins
Don't use the root folder, but your own folder in .config (or Users).

This worked for me, but this file doesn't seem to contain as many items as the flatpak had.

Filters > Enhance > Heal Selection
Filters > Enhance > Heal Transparency
Filters > Map > Resynthesize

I also found another zip file called GIMP-Extras.zip which contained the following, but not all of them worked:

Map > Style
Render > Texture
Enhance > Enlarge & Sharpen
(this one is obsolete)
Fill Pattern (Not in the Filter menu)
Enhance > Uncrop
Resynthesis > Sharpen
(this one is obsolete)

Since they don't all work, and I don't remember where I got them, in the interest of safety, I'm not going to provide a link.

So let's use the first three.

Heal Selection is the one I covered three years ago. You can use a selection tool to select the item you want to remove, then select **Filters > Enhance > Heal Selection**. Whatever you selected should disappear in a few seconds. Here I have a photo of a mountain scene I took on a vacation years ago (right, top). I want to use it, but there's an ATV on the left side of the photo that I want to remove without removing any of the ruins that are there. Using a selection tool, I drew around the ATV, then selected **Filters > Enhance > Heal Selection**.

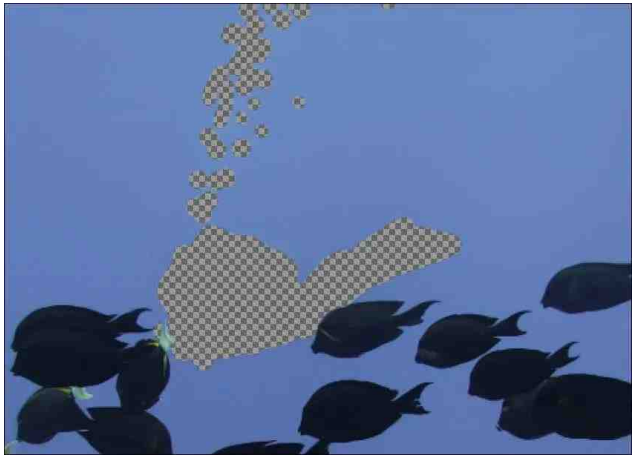
You don't have to be super careful around items that are staying, unless they are right next to something you're deleting, so I went around the ATV loosely, except for the piece of wood I wanted to keep. Make sure your selection curve is closed (go all the way around, even along the edge, and when you click on your beginning, press the Enter key to close your selection.

You'll get a small window asking how big your context sampling needs to be (how far out from your selection you want to use) and where you want to take the sample from. The filter takes pixels from around the selection and fills in the spot being removed. The default is 50.

Here's the result, and the ATV is gone.

Heal Transparency is another method to remove something. You can load your photo, add an alpha channel (for transparency) if it doesn't already have one, and use your eraser tool to remove what you want. MAKE SURE your eraser has a sharp edge, because if you use a fuzzy edge, it won't work.

Here I had a photo of a scuba diver underwater. I decided to remove the diver and leave the fish. Using my eraser tool, I removed the diver and the bubbles, and then chose **Filters > Enhance > Heal Transparency**.



It worked, but I thought I could see some white areas up at the top.

When I zoomed in, I saw some impressions that looked like my eraser strokes on the air bubbles.

I can only conclude that your eraser strokes need to be continuous rather than just spots. When I undid my work and redid it, I got something better.

Scuba diver photo from Wikimedia Commons

Map > Resynthesize is the third plug-in in the original package. It is the hardest. The other filters are specific directions to do one thing, but this is the controller for everything it can do.

From Github's wiki:

The hardest to understand, but the most powerful. Displays every control of the Resynthesizer engine plugin, then runs it.

Seems like a "control panel." Here, you can experiment with other uses of the algorithm.

To use Map > Resynthesize, you usually:
• first create other layers and select areas in them,
• start Map > Resynthesize,
• choose those layers in the dialog,
• choose other parameters,
• finally choose OK to run.

But you must know specific steps to get one of the many effects. That's what the other plugins do automatically, they "know" the steps. The other plugins can all be replicated by specific steps using Map > Resynthesize.

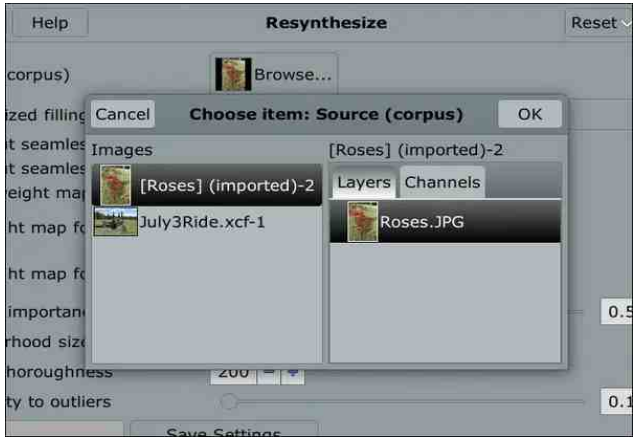
One obscure use of Map > Resynthesize is to make seamlessly tillable tiles. If you tile the plane with such tiles, you can still see that there are rows and columns, but there are no hard edges between the tiles. (Future: make this use another plugin.)

One of the things you can do is replace a part of one image with a part of another. I went back to the mountain scenery where I removed the ATV. I decided to replace the metal box on the other side of the ruined building with some roses, so I loaded the original picture and also a photo of some roses;

The first thing I did was resize the roses so they wouldn't look like giant flowers in the other photo. I selected the part I wanted to replace in the first photo, and the flowers I wanted to replace it with from the second photo, then chose **Filters > Map > Resynthesize**.

The window you get looks like this:

If you have layers in your project, the Source setting will let you browse to each layer so you can use part of it. I loaded each picture into GIMP separately, so the Source selection button will show each image I have loaded. The left side of the Source window shows what's available, and the right side shows what layers you can use (if you have layers). Choose what you want to use on the left, and choose from the layers on the right, then click OK.



I left all the other defaults the first time.

It's not perfect, but you can see what the filter does. I'll have to unselect the rose bush and clean up around it before I'm finished. I'm sure that with some practice, you can do something really cool.

I'm still trying to figure out the tiling feature, so I'll cover it later.

The PCLinuxOS Magazine

Created with Scribus

Screenshot Showcase

Posted by mutse, on June 2, 2026, running Mate.

A New Script To Create Image Transparency

by Paul Arnote (parnote)

It's definitely no secret: I spend a LOT of time working with graphic files in my position as the chief editor for The PCLinuxOS Magazine. You would think that with as much time as I spend working with graphic files, I'd be some kind of GIMP savant … but I'm not. My GIMP skills, if I had to rate them, are just a hair above average (my own assessment). On the staff of The PCLinuxOS Magazine, Meemaw is the resident graphics guru.

One ability/skill I *wish* I had would be to open up an image in GIMP and remove the background from the image, replacing that background with transparency. Even today, all these years later, if I'm able to perform this task, it happens entirely by accident (and not because I know how to accomplish it). More times than not, I end up sending that image to Meemaw, and have her email the image back to me with the transparent image as an attachment.

And, honestly, I'm sure I could "learn" this ability, if I applied myself a little more towards learning it, but I have so many other things to keep track of and that require my attention … both magazine related and non-magazine related things. So, it's a learning opportunity that I've let fall by the wayside because of other commitments that have a higher priority. Hey … at least I admit it and own it.

Image by Julien Tromeur from Pixabay

While I'm not a GIMP Super User (I'm happy to let Meemaw take that title … and happy that we have her), I do know quite a bit more than the average bear when it comes to graphics file formats, their capabilities, and their shortcomings. That knowledge was extremely helpful in creating this new script.

So many times over my time as the chief editor of The PCLinuxOS Magazine, I can't even begin to count how many times I've needed or wanted an image with a transparent background. In the end, I would always find a way to work around the issue (or I just sent the image to Meemaw to work her magic when I couldn't find a workaround).

A Little History

So, you might be wondering why or how this whole situation unfolded and came about. Way back in March 2013, I wrote a couple of scripts, and published them in The PCLinuxOS Magazine. One was called "convert-image.sh" and the other was called "img-resize.sh." Then, in February 2024, I revisited and updated the img-resize script. In March 2023, I also updated the convert-image script.

The truth of the matter is that I've made multiple minor updates to these two scripts several times over the years. They are honestly two scripts that Meemaw and I use often when we lay out articles for the magazine's PDF editions. Like. All. The. Time. Since we've already run these scripts twice in the magazine, I haven't bothered to post or publish updated versions of them.

But, somewhere along the line, something in ImageMagick (whose tools perform the heavy lifting in these scripts) changed. One of the things that changed is that sometimes, the scripts will produce an image with a black background, instead of a transparent background as it was expected to do.

So, recently, I set out to "fix" the scripts. I discovered that ImageMagick has become quite "picky" about the order of the commands issued, much to my dismay. If you've ever read the "usage" "rules" for the ImageMagick commands, you already know how utterly confusing those rules are. They tell you everything you need to know about all of the commands and all of the options … except the order they need to be executed in. If they do divulge that information, I've not been able to find it in the literal mountain of information dispensed. I liken finding things on their site to trying to find a single bobby pin in an acre patch of thick weeds.

Unable to find an option or sequence of commands that worked like I needed it to, I took a different approach. I created a new script that only works to produce transparency in an image. I call that script **remove-bg2.sh** (remove-bg.sh, the initial version, was just a "proof of concept" script). While "fixing" the original scripts is still definitely on my radar, this new script fills my needs for now.

A Brief Note About Graphic File Formats

Of all the graphics file formats out there, there are only six major formats that support transparency. We'll immediately eliminate two of them from consideration right from the start: GIF and HEIC. GIF files only support one-bit transparency, so it really isn't of much use to us (plus it has other significant issues that are beyond the scope of this article). HEIC files are primarily used on Apple devices, and (despite its benefits) it hasn't gained much traction outside of the Apple Universe.

The rest of the other major graphics file formats out there … JPG (along with ***most*** of its variants), BMP, and most of the others … do not support transparency.

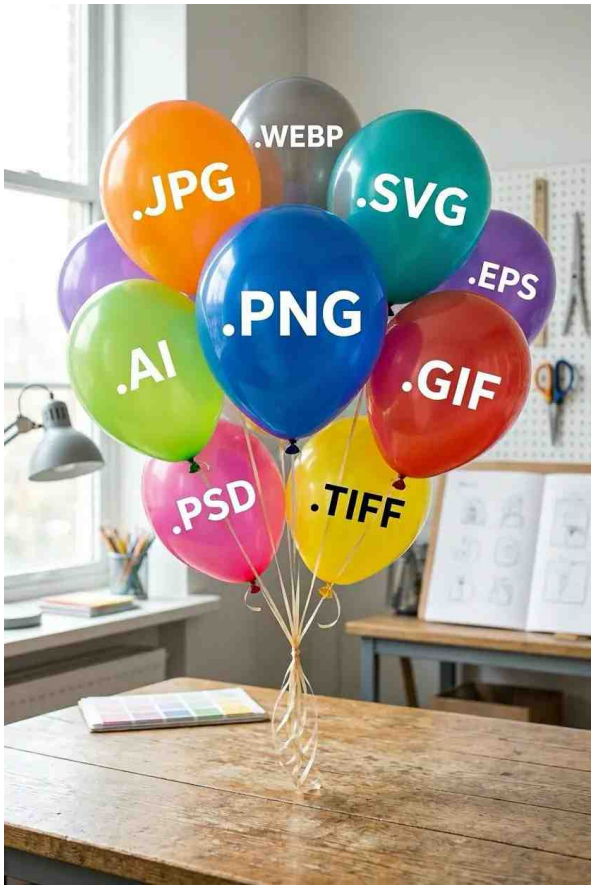


Image by Bing Image Creator

That leaves only four graphic formats that we really have to worry about: PNG, WEBP, AVIF, and TIFF. While I've included TIFF as one of the output formats, I can't see that a lot of people will opt to use it, due to the fact that it produces VERY large file sizes. Still, I included it because it supports transparency. We're all familiar with PNG files, and WEBP graphics are quickly gaining in popularity. In fact, we've been using WEBP graphics for the magazine PDF and HTML for more than two years, thanks to its great quality and vastly smaller file sizes. The outlier here is AVIF, which, while a quality graphic format, hasn't gained as much widespread acceptance as WEBP graphics. AVIF files definitely rival WEBP for producing high quality images with significantly smaller file sizes. The biggest reason we don't use AVIF graphics here in the magazine is that Scribus (the program we use to produce the magazine's PDF edition) doesn't yet support their use. If or when Scribus does include support for AVIF graphics, we'll be open to

using them. All of the "major" web browsers already support AVIF graphics, so we're not averse to using them in the HTML edition.

So, for now (and for our uses in The PCLinuxOS Magazine), we're pretty much relegated to using only PNG and WEBP graphics files when we need images that include transparency.

Let's Be Transparent

This script, like several other scripts I've written over the years, is designed to be run either from the command line, or as a Thunar Custom Action. I'm certain that users of other file managers can also use this script in a similar manner. However, since I'm strictly a Xfce user, the exact process for doing so is not something I'm intimately familiar with.

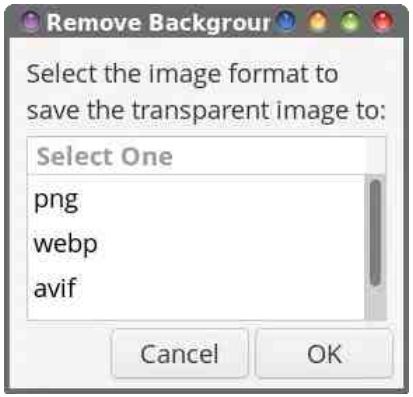
You can, of course, download the script from the magazine server. The file name is "remove-bg2.sh.txt," and it's only 3.2 KiB in size. As I've mentioned before with other scripts that I've presented in the magazine, store the script in the directory where you normally save your bash scripts. Remove the ".txt" file extension, and make the file executable. Hopefully, the directory where you save your bash scripts is in your computer's $PATH statement. If it is, all you then have to do is issue the script name (remove-bg2.sh) to start the script. Otherwise, you will have to provide a full path to the script to run it.

So, before we talk about the script, let's take a look at it.

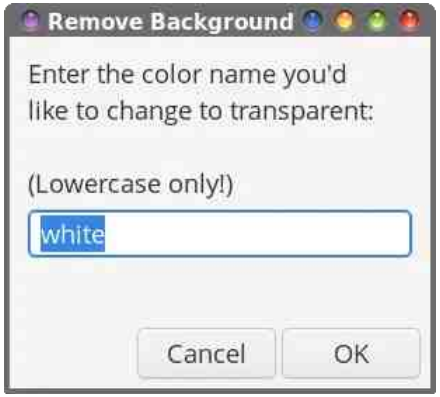
```
#! /bin/bash
#
# Written by Paul Arnote, Chief Editor of The PCLinuxOS Magazine
# Read the original article in the July 2026 issue of The PCLinuxOS
# Magazine. https://pclosmag.com/html/Issues/202607/links.html
#
# Released under the GPL 2.0 license
# May be freely distributed in accordance to the GPL 2.0 license
#
# This script will take an image and change the specified color
# to transparent, and then resave that image to one of the four
# graphic formats that support transparency (PNG, WEBP, AVIF, or TIFF).
#
# This software is offered without warranty of any kind.
# Any use is at your own risk.
#
# Usage: remove-bg2.sh [name of file]
#
# The new image will be saved in the same directory as the original,
# without any risk of overwriting the original file. Only ONE file
# at a time can be processed.
#
# This script will also function as a Thunar Custom Action. The
# command line for the custom action should be remove-bg2.sh %n.
# Under "Appearance Conditions," place a checkmark in front of
# "Image Files," and leave the file pattern set to *.

# Select the image format you want to save the image as
EXT=$(zenity --list --column="Select One" --title="Remove Background"
--width=250 --height=250 --text="Select the image format to\nsave the
transparent image to:" png webp avif tiff)
# If the 'Cancel' button is selected, exit the script
    if [ $? == 1 ]; then
       exit
    fi
# Make sure the file extension is all lowercase text, and save it
declare -l EXT
EXT=$EXT

# Enter the color name or hexadecimal equivalent. To see a list of
# acceptable color names and their hexadecimal equivalents by visiting
# https://imagemagick.org/color/#color_names
# The color defaults to "white." The other commonly used color will
# most likely be "black."
COLOR=$(zenity --entry --title="Remove Background" --width=300 --
height=250 --text="Enter the color name you'd\nlike to change to
transparent:\n\n(Lowercase only!)" --entry-text="white")
# If the 'Cancel' button is selected, exit the script
```

```
45.    if [ $? == 1 ]; then
46.       exit
47.    fi
48.
49. FUZZ=$(zenity --entry --title="Remove Background" --width=300 --
height=250 --text="Enter the fuzz value you'd\nlike to use.\n\nDefault:
10\nUse smaller numbers only large\nenough to achieve the results\nyou are
seeking!" --entry-text="10")
50. # If the 'Cancel' button is selected, exit the script
51.    if [ $? == 1 ]; then
52.       exit
53.    fi
54.
55. # Make sure the input file actually exists
56. if [ ! -e $1 ]; then
57.     continue
58. fi
59.
60. # Strip away the file extension of the input file, and save it in
61. # the variable "name"
62.     name=$( echo $1 | cut -f1 -d.)
63.
64. # The ImageMagick "convert" command performs the creation of the
65. # image with transparency. This command is based on the solution
66. # presented by fmw42 on Stack Overflow, incorporating the alteration
67. # offered by mobeen, along with some of my own tweaks.
68. # https://stackoverflow.com/questions/69851329/ \
69. # remove-backgroundany-color-from-image-using-image-magick
70.     convert $1 -alpha off -fuzz $FUZZ% -fill none -transparent $COLOR
-draw "alpha 0,0 floodfill"  \
71.     \( +clone -alpha extract -blur 0x2 -level 50x100% \) \
72.     -alpha off -compose copy_opacity -composite \
73.     $name-fuzz-$FUZZ-transp.$EXT
74.
75. # Exit the script cleanly after the creation of the transparent image
76. exit 0
```

The script starts off in **line 1** with the typical shebang (#! /bin/bash) that typically starts all bash scripts. The next 25 lines are comment lines, which should be self-explanatory.



Lines 29 - 36 display the first Zenity dialog box that allows the user to select the output format for our image with transparency. If the "Cancel" button is selected in the Zenity dialog box, the script exits gracefully. To further ensure that things are as we need, we make sure that the output format file extension is all in lowercase lettering.

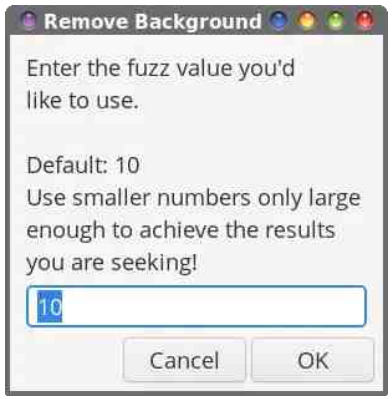


In **lines 43 - 47**, the second Zenity dialog box allows the user to specify the color that we want to replace with transparency. The default value is "white," but can be changed to whatever color you want or need. If using the color ***names***, be sure to keep them all lowercase letters. You can also enter the six-digit hexadecimal RGB color designation (e.g., #FFFFFF for white, or #000000 for black), instead of the color name. You can find both

the color names and their six-digit hexadecimal RGB color designations at the ImageMagick website.

Probably the most common colors you'll need are either white (the default) or black. As for figuring out what the other colors are, you could just open the image in GIMP and use the eyedropper to copy the color you're interested in. Then, click on that color in the GIMP foreground/background color control. Pay attention to the six-digit hexadecimal code for that specific color, and use that.

So, if you're opening the image up in GIMP to get the color, you may be asking "why not just use GIMP to change that color to transparency?" Well, if you're like me, you might not yet have figured out that entire process (I know I haven't fully figured it out yet). So, instead of stumbling and fumbling around in GIMP trying to figure that entire process out (and ***maybe*** accidentally figuring out the necessary steps), just grab the color code and return to the script, which will handle it for you. There's no doubt that GIMP can handle the task, but the script is quicker.



We create another Zenity input dialog in **lines 49 - 53** to allow the end user to specify the "fuzz" level (for comparison, it's called "threshold" in GIMP). The fuzz option allows for a variance in the color you specified, to help make sure there are no stray pixels left in the area you want to be transparent. In the script, we set the default value to 10%. Setting it to 0% means that ONLY the exact color you specified is changed to transparency. With the 10% setting, that means that the color you specified and other colors within 10% of that color are changed to be transparent. I have found that the 10% setting tends to work fairly well for my needs in most cases, which is why I selected it to be the "default" fuzz level. Don't be afraid to play around with this setting to achieve the results you are seeking. Resist the "urge" to use big numbers! You most likely won't like the results. Typical fuzz levels are between 4 and 10 percent. Once you get much above 30 percent, too much of the original image is replaced with transparency to even be recognizable or usable.

So, here's an example of how different fuzz levels affect the image that's output. I selected a random image with a mostly monochrome background. In fact, that image appears elsewhere in this issue, in the ICYMI article.

Original image:

Image by Satheesh Sankaran from Pixabay

Now, here's a composite image showing the effects of different "fuzz" levels:

When specifying a color, I just "winged it" (I also call it a WAG … a wild a** guess) and entered "blue" as the color I want to replace with transparency. With a fuzz level of 5% (upper left), you can see some of the blue background being replaced with transparency. A fuzz level of 10% (upper right), even more of the blue background has been replaced with transparency. Using a fuzz level of 20% (lower left), we've almost completely converted all of the blue background to transparent, except for the small area under the "weight loss" text and to the right of the scale. With a fuzz level of 30% (lower right), we've completely eliminated all of the blue background, replacing it with transparency.

Just to demonstrate how you should keep the fuzz level as small as you need to achieve the results you need, here's the same image with a 50% fuzz level:

As you can see, we surpassed the threshold of where only the specified background color is converted to transparency. Some of the blue reappears, and the dark gray pad of the scale has now been converted to transparency. Some of the yellow text banners have also been converted to transparency, along with the shadow area along the right side of the scale. So, it's important to use as small of a number in the fuzz level as possible to achieve your desired results. If you use too big of a number in the fuzz level, you start, in essence, regressing and producing diminishing returns from what we were seeking. ***Even a one or two percent change can make a huge difference.*** Do try multiple fuzz levels to find that level that gives you just what you need. In the examples above, a fuzz level of 30% works, while a fuzz level of 50% does not.

Lines 56 - 58 double check to make sure that the input file actually exists. If it does, the script is allowed to continue. **Line 62** strips the file extension from the input file, and stores it in the $name variable.

The "heavy lifting" in this script is done by **lines 70 - 73**, invoking the ImageMagick **convert** command. We use the $FUZZ variable to insert the fuzz level that we literally just talked about. In the rest of the convert command, the color we specified earlier is used (-transparent $COLOR) to set which color we want to change to transparency. Then, we use the input file name (minus its extension) that we saved into the $name variable, append the word "-fuzz-" and the fuzz level ($FUZZ) to the name, append "-transp" to the file name, and give it the new file extension we selected in the first Zenity dialog box. This helps to ensure that your original file is ***not*** overwritten (I figure it's better to be safe than sorry!). It also saves images made with different fuzz levels to separate files so you can monitor and see how well each fuzz level works for your specific image. This ImageMagick "convert" command is based on the solution presented by **fmw42** on Stack Overflow, incorporating the alteration offered by **mobeen**, along with some of my own tweaks.

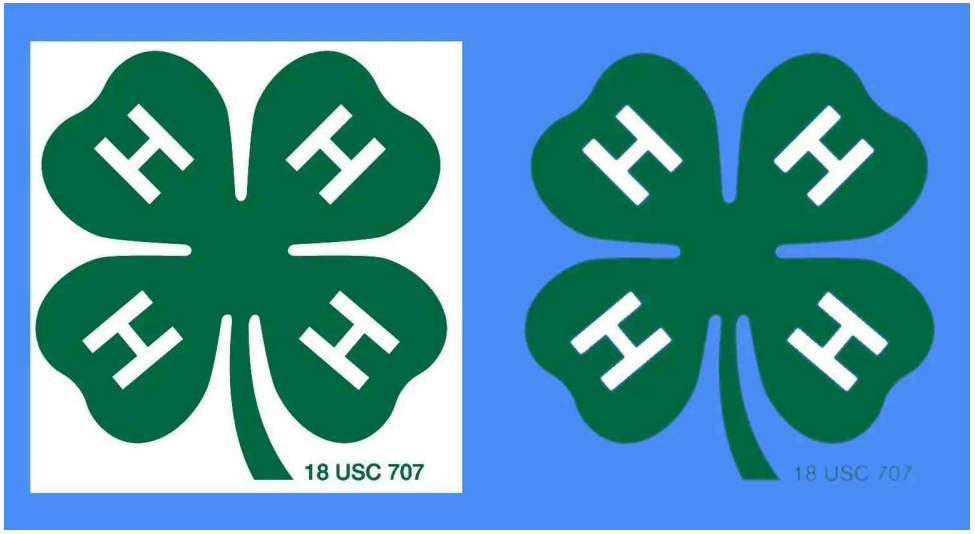
Finally, in **line 76** of the script, we allow the script to exit cleanly and gracefully after the image is converted.

A Couple of Other Examples

So, let's take a look at some other examples, some of which were created just for this article.

This composite image shows the JPG image downloaded from Pixabay on the left (used earlier in this article), and the PNG with the background converted to transparency on the right. The images were placed on a gradient blue background to assist with displaying the transparency. The image on the left is the JPG (and remember, JPG files do NOT support transparency), while the image on the right is the PNG created after processing with the script. In my opinion, I think the script did a very good job with this image. The original JPG has an all white background, and the PNG file has no background. This image was a little tricky, and required a fuzz level of only 1%. Anything more would make transparent "holes" in his chest plate, teeth, and the whites of his eyes. Had I been able to use a higher fuzz level, the white surrounding the character would have been eliminated (and it was eliminated in my initial attempts at higher fuzz

levels). So, as long as I reproduce the image on something that also uses a white background (like the pages of The PCLinuxOS Magazine), no one will ever notice the white "halo" around the image.



Meemaw contributed this image on one of her first tries using the script. The JPG file is on the left, with a solid white background. The image on the right is the transparent image created by the script. In removing the white background, the "H" letters on the clover leafs were also converted to transparent, but Meemaw went into GIMP and refilled them with white. She used the "default" fuzz level of 10% when creating the transparent image. We're showing it on a blue background, so you can better discern the transparency of the image on the right, versus the solid background of the JPG on the left.

Using remove-bg2.sh

The remove-bg2.sh script takes only one command line argument, and that is the filename of a graphics file you want to add transparency to. It can be any graphics format that ImageMagick supports (meaning, that it can read, and that's a LOT of different graphic formats!). Look for the ones in the list marked "R" under the Mode column.

From the command line, that will look like this:

```
remove-bg2.sh [name-of-file]
```

or

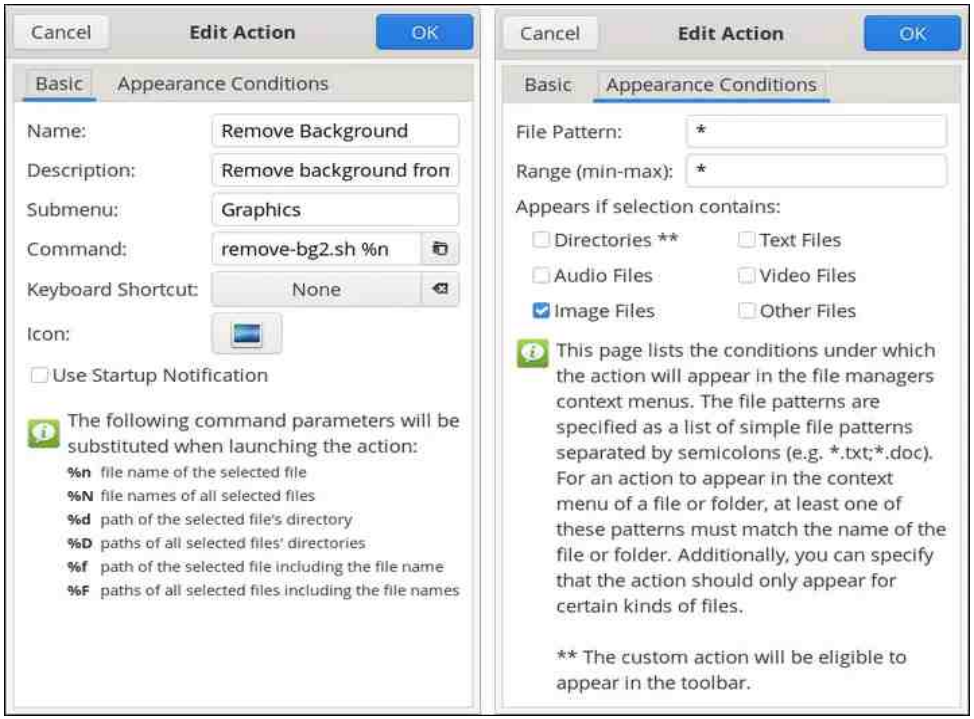
```
remove-bg2.sh [path-and-name-of-file]
```

Keep in mind that this script will ONLY work on ***ONE IMAGE FILE AT A TIME!*** That remains true whether you're using it from the command line or from a Thunar Custom Action.

You should ensure that filenames **DO NOT** have spaces within them. To keep things simple, I've not added anything to the script to strip out spaces in the input filenames (I tried, and wasn't successful … just changing the spaces in a filename is not enough within the script, since options in the convert command rely on the specified image actually existing on your hard drive, and not just the altered name of the file held in the script's memory/variable … and none of my usual "tricks" work). Spaces in filenames **WILL** cause the script to fail, **so you have been warned**. And, since it's working so well, I've opted (at this time) to not go in and mess with things to make sure filenames with spaces don't cause a script failure. Spaces in filenames really is a horrific idea anyways, as spaces in the Linux command line (and thus, bash scripts) are overloaded, and are VERY frequently used to separate one set of command line options from another.

I tend to avoid spaces in filenames as a hard rule, having been bitten in the backside numerous times from having spaces in filenames. I have special Thunar Custom Actions that change spaces in filenames to either "-", ".", or "_" (that's a dash, period, or underscore), depending on which custom action I choose to use. So, if I download a file that contains spaces in its filename, I immediately strip those spaces from that filename with one of those custom actions. It's just something that I routinely do.

Using remove-bg2.sh As A Thunar Custom Action



In the "Edit Action" dialog box, under the "Basic" tab, enter "Remove Background" in the name field. In the "Description" field, I've entered "Remove background from image." If you use submenus in your Thunar Custom Actions, enter the name of the submenu you want the custom action to appear under. On my computer, my graphic custom actions are grouped under the "Graphics" submenu. Then, enter "remove-bg2.sh %n" in the "Command" field. I don't think I've ever set a keyboard shortcut for any of my custom actions … ever. But, I do choose and set an icon to be displayed next to my custom action in the menu.

Under the "Appearance Conditions" tab, leave the * in the "File Pattern" field, and place a checkmark in front of "Image Files."

Now, your custom action will be ready to use on the image of your choice. Find the image, right-click your mouse on it, and select the custom action. Follow the prompts from the script, and you'll be producing your own images with transparency. Again, keep in mind that the script is designed to work with only ***one image at a time*** (hence the use of %n in the command entry line).

Alternatives

Well, of course you can work in GIMP and try to learn the process there to create images with transparency … if you have the time. I haven't had that time available thus far.

Meemaw, in her bimonthly GIMP tutorials, wrote two separate tutorials on how to get rid of the background in images (that is, create transparent backgrounds) back in 2021. If you have an image that you want to have a transparent background, you could follow along with her well-written tutorials to achieve your desired results. Only one of her tutorials relies on the use of masks. For some reason, I have a real mental block when it comes to the use of masks in GIMP, so I have to tap out on that one. Fortunately, she offers three other ways to create a transparent background.

Or, you can also head on over to the https://remove.bg website, upload your image, and then download the image with a transparent background back to your computer. While that whole process takes considerably longer than running this script, you ***might*** obtain better results than from the script. See, that website uses either AI or some complex algorithms (that are WAY beyond my abilities to code) to produce the image with a transparent background. As a result, it can handle much more complex images that you might want to have a transparent background in than this script can handle. But then, keep in mind that you're uploading your files to some distant website, so there's an ever-present question of security. With the remove-bg2 script, everything is staying right on your computer.

Summary

I hope you find this script worthy of gaining a place in your toolbox full of graphics tools. And, I have to admit that creating this script was fun (yeah, I know … I have a warped sense of fun, huh?). But part of the "fun" was discovering how much more capable this script is than I had originally thought. Initially, I wasn't going to include an option to adjust or specify the fuzz level, and just leave it at the default value of 10% hardcoded into the script. But, once I added the ability to make the fuzz level customizable, it opened up far more capabilities than I could have ever imagined, and made the script work much better than I had hoped.

Is this script perfect? Nope. Does this script deliver "perfect" results? Not always. But, the results are close enough to perfect for my use. At the very most, all you should have to do is to open the output image in GIMP and perform some minor tweaks. And THAT process is far faster and more secure than uploading your images to some far off, distant website.

PCLinuxOS

Users Don't
Text
Phone
Web Surf
Facebook
Tweet
Instagram
Video
Take Pictures
Email
Chat
While Driving.

Put Down Your
Phone & Arrive
Alive.

PCLinuxOS Magazine Graphics Special Edition, Volumes 1 - 4

Uhleash your GIMP & Inkscape skills. Over 160 tutorials. Grab your FREE copy now!

Screenshot Showcase

Posted by parnote, on June 16, 2026, running Xfce.

Wiki Pick: Getting Rid Of Unwanted/Unneeded Files

Relevant to all versions of PCLinuxOS

This page is the results of the questions asked about how to remove unneeded/unwanted files on the PCLinuxOS forum.

This **ONLY** involves cleaning old logs and other system stuff that is not needed.

Removing Unwanted/Unneeded files

These files can be found in the /var directory and are log and cache files. There are several ways to get rid of these unwanted and unneeded files.

The first is to remove them by hand.

Open a console window and su to root

Now you can simply enter the following lines one at a time pressing enter after each line.

```
rm -rf /var/cache/cups/job*
rm -rf /var/cache/fontconfig/*
rm -rf $(find /var/spool -type f)
rm -rf $(find /var/lib/spool -type f)
rm -f $(find /var/log -type f -iname '*.old')
rm -f $(find /var/log -type f -name '*.gz')
rm -f $(find /var/log -type f -iname '*.[123456789]')
cat /dev/null |tee $(find /var/log -type f  -iname '*log')
cat /dev/null |tee /var/log/dmesg
cat /dev/null |tee /var/log/explanations
cat /dev/null |tee /var/log/messages
cat /dev/null |tee /var/log/wtmp
cat /dev/null |tee /var/log/ConsoleKit/history
```

***DO NOT** just simply delete files in your /var/log folder, as some of these files are indeed needed by the system while it is running.*

This is a list of some of the files that are required by the system while running.

```
dmesg, explanations, messages, wtmp and history.
```

You will note in the above list, they start with the command **cat /dev/null >**.

This command **DOES NOT** delete the file, it simply "**empties**" the contents of the file. Again **DON'T** delete these files.

The above is not only time-consuming and error-prone, but it takes a lot of time to enter and execute each and every line.

Image by Olena from Pixabay

Forum user **footstep11** brings a bit better solution by saving the commands to a filename called docleaning. To do this, simply copy the following lines and save them in your home directory as a file called **docleaning**.

```
rm -rf /var/cache/cups/job*
rm -rf /var/cache/fontconfig/*
rm -rf $(find /var/spool -type f)
rm -rf $(find /var/lib/spool -type f)
rm -f $(find /var/log -type f -iname '*.old')
rm -f $(find /var/log -type f -name '*.gz')
rm -f $(find /var/log -type f -iname '*.[123456789]')
cat /dev/null |tee $(find /var/log -type f  -iname '*log')
cat /dev/null |tee /var/log/dmesg
cat /dev/null |tee /var/log/explanations
cat /dev/null |tee /var/log/messages
cat /dev/null |tee /var/log/wtmp
cat /dev/null |tee /var/log/ConsoleKit/history
```

Now to run this file, open a console window and su to root. At the prompt, type in the following command and then press enter

```
sh ./docleaning
```

While this is somewhat better, you still have to open a console, change to the root user and type in some code. I don't know about you, but I just don't remember all of the possible commands that are required all the time.

Editor's Note: *You can take this one step further, and create a bona fide bash script that checks to make sure the user is launching it as the root user (and exits if not), and then executes the commands. Then, you can also build a .desktop file for it, and place that desktop file on your desktop or in a launcher on your panel. That way, all you have to do is to enter the root password when prompted.*

Here's one possibility that I threw together in less than a couple of minutes:

```
#! /bin/sh

# MUST BE RUN AS ROOT USER
if [ "$UID" != "0" ]; then
  zenity --warning --width=275 --title "Privilege Error" --text "You must
run this program as the root user."
  exit 0
fi

rm -rf /var/cache/cups/job*
rm -rf /var/cache/fontconfig/*
rm -rf $(find /var/spool -type f)
```

```
rm -rf $(find /var/lib/spool -type f)
rm -f $(find /var/log -type f -iname '*.old')
rm -f $(find /var/log -type f -name '*.gz')
rm -f $(find /var/log -type f -iname '*.[123456789]')
cat /dev/null |tee $(find /var/log -type f  -iname '*log')
cat /dev/null |tee /var/log/dmesg
cat /dev/null |tee /var/log/explanations
cat /dev/null |tee /var/log/messages
cat /dev/null |tee /var/log/wtmp
cat /dev/null |tee /var/log/ConsoleKit/history

zenity --info --width=275 --title “Docleaning” --text “Docleaning has
finished cleaning up unwanted/unneeded files.”

exit 0
```

The script checks to see if you’re the root user. If not, it exits before any of the commands are executed. The command to start the script should read something like “pkexec docleaning.sh”, provided that you store the script in a directory that is in your system’s $PATH statement. Otherwise, the command to start the script will most likely read something like “pkexec [path-to-script]/docleaning.sh”.

Now comes CRON to the rescue

The nice part is we only need to set it up once, and then let cron do all the work, and we can forget about it.

Here's what we need to do to set it up.

Copy the following lines:

```
@daily rm -rf /var/cache/cups/job*
@daily rm -rf /var/cache/fontconfig/*
@daily rm -rf $(find /var/spool -type f)
@daily rm -rf $(find /var/lib/spool -type f)
@daily rm -f $(find /var/log -type f -iname '*.old')
@daily rm -f $(find /var/log -type f -name '*.gz')
@daily rm -f $(find /var/log -type f -iname '*.[123456789]')
@daily cat /dev/null |tee $(find /var/log -type f  -iname '*log') && cat /
dev/null |tee /var/log/dmesg && cat /dev/null |tee
/var/log/explanations
@daily cat /dev/null |tee /var/log/messages && cat /dev/null |tee /var/log/
wtmp && cat /dev/null |tee /var/log/ConsoleKit/history
```

*(**Tip within a tip**: each new line in the cron job listing starts with @daily. If it starts with @daily, then that signifies a new line in the file. This will help discern each individual line, and not where the text is wrapped. You can also replace @daily with @weekly, or any other timeframe recognized by cron.)*

Save them as a file called root to /var/spool/cron. **You will need to be the root user** to save this new file.

1. Open a console window, become the root user.

2. Open nano (IE: nano /var/spool/cron/root)

3. Paste the above lines into nano.

4. To save the file, press Ctrl + x.

5. Answer “Y” when asked to Save modified buffers. Then press enter.

Your new file has been saved. You will need to set the permissions for your new file. Again as root in the console window, type in the following at the command prompt:

```
chmod 600 /var/spool/cron/root
```

Press enter. That's it. You can now close the console window.

Now cron will do the job of cleaning your files automatically, daily.

You can view this PCLinuxOS Knowledgebase Wiki article here.

Tip Top Tips: KDE 6.x.y Change Shutdown/ Logout Countdown (Konsole Method)

Editor's Note: *Tip Top Tips is a semi-monthly column in The PCLinuxOS Magazine. Periodically, we will feature – and possibly even expand upon – one tip from the PCLinuxOS forum. The magazine will not accept independent tip submissions specifically intended for inclusion in the Tip Top Tips column. Rather, if you have a tip, share it in the PCLinuxOS forum's "Tips & Tricks" section. Occasionally, we may run a "tip" posted elsewhere in the PCLinuxOS forum. Either way, share your tip in the forum, and it just may be selected for publication in The PCLinuxOS Magazine.*

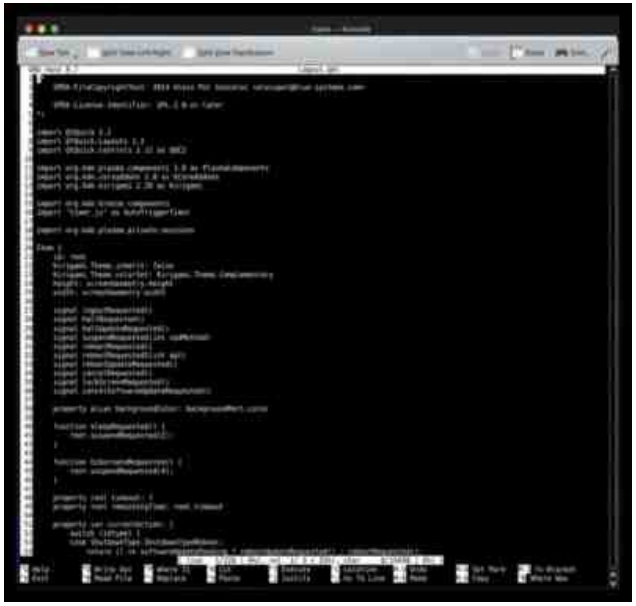
This month's tip comes from **keltonix**.

You can change the shutdown/logout countdown timer for KDE 6.x.y using a couple different methods. The following method uses **Konsole** and the **nano** text editor:

First, open a Konsole window, type **su** and press the Enter key. Enter your root password when prompted.

Then, type **cd /usr/share/plasma/look-and-feel/org.kde.breeze.desktop/contents/logout**, and press the Enter key.

Next, type **nano Logout.qml** on the command line prompt. The preceding actions will open the file **Logout.qml** in the **nano** text editor, similar to the image below.



In the **nano** text editor, press the keyboard combination **Ctrl+w**. Type **property real timeout**, and press the Enter key. You will be magically taken to the line that reads **property real timeout: 30**. Change the **30** (*which in this case refers to 30 seconds*) to a length of time in seconds that you prefer, (e.g., **7**).

Press and release the keyboard combination, **Ctrl+o**, and press the Enter key. Then, press and release the keyboard combination **Ctrl+x**, and press the Enter key.

The preceding actions have edited the file **/usr/share/plasma/look-and-feel/org.kde.breeze.desktop/contents/logout/Logout.qml**, saved the changes, and then exited the **nano** text editor.

Now you can type **exit** to exit the root user's profile in Konsole, and press the Enter key. Then, you can type **exit** to close the Konsole window, and press the Enter key.

All done. Next time you shut down your computer, the new countdown timer setting will be in effect.

Ramchu, in his reply, found a slightly different solution to the logout/shutdown delay.

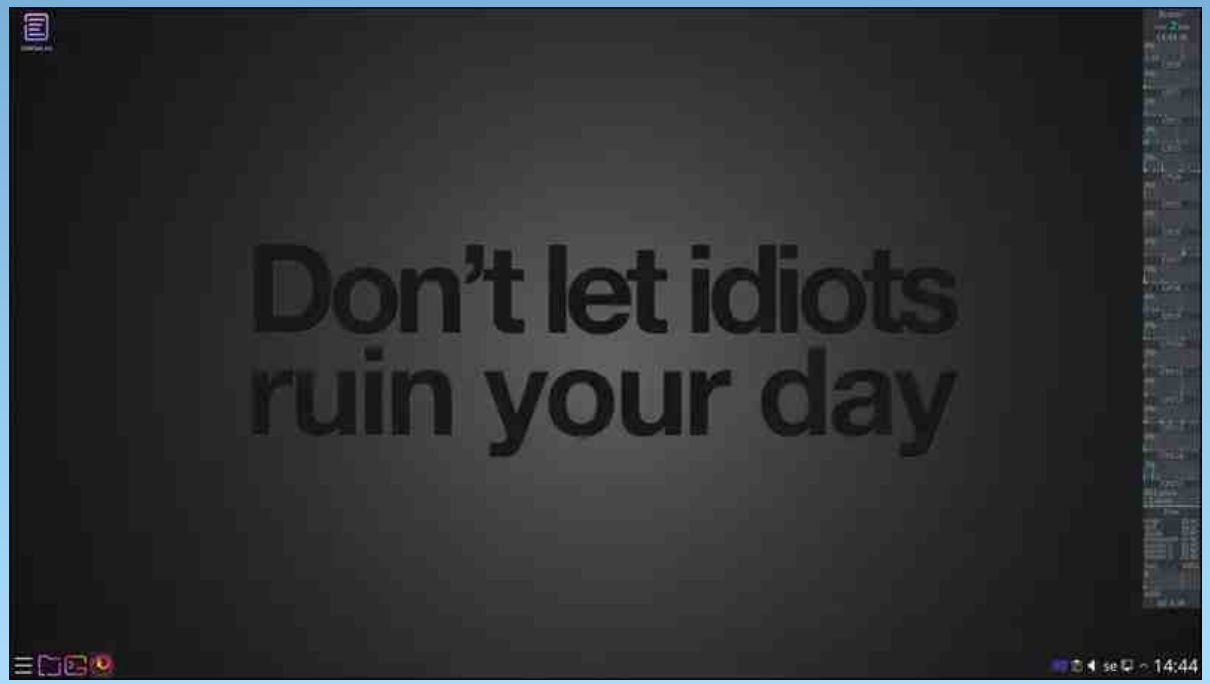
The instructions that you have posted are good information to know, but for myself, I just went to: System Settings> Session> Desktop

Session> Ask for Conformation: unchecked On shutdown, restart, logout. Now I have no timer involved when I shutdown, restart or logout, it just does it. I have always found that second confirmation click to be annoying and needless as I have already made my decision. Has this caused me a few problems, the answer is yes, but nothing that I could not recover from.

Donate To PCLinuxOS

Community Supported.
No Billionaires/Millionaires.
No Corporate Backing Or Funding.
Click here to make a one-time donation through Google Checkout.
Or, click one of the amounts down below to make a monthly, recurring donation.

Screenshot Showcase



Posted by scoundrel, on June 2, 2026, running KDE.

PCLinuxOS Recipe Corner Bonus

Meatloaf Patties

Serves: 4

INGREDIENTS:

1 egg, lightly beaten
1 slice soft bread, blended into crumbs
½ teaspoon salt
½ teaspoon ground black pepper
1 pound lean ground beef
1 (8 ounce) can tomato sauce
3 green onions, chopped
2 ¼ tablespoons brown sugar, or to taste
2 teaspoons Worcestershire sauce
2 teaspoons prepared yellow mustard

DIRECTIONS:

Mix egg, bread crumbs, salt, and black pepper together in a large bowl. Break ground beef into small pieces and add to the bowl; mix well with your hands until beef is evenly moistened. Shape beef mixture into four patties.

Heat a skillet over medium heat. Cook patties in hot skillet until completely browned, 2 to 3 minutes per side; remove to a platter and drain grease from skillet.

Return the skillet to medium heat. Stir tomato sauce, green onions, brown sugar, Worcestershire sauce, and yellow mustard together in the skillet. Gently add patties into the sauce and bring to a boil; reduce heat to medium-low and simmer until patties are very firm, hot, and cooked through, about 10 minutes more. An instant-read thermometer inserted into the center should read 160 degrees F (70 degrees C).

NUTRITION:

Calories: 302 Carbs: 15G Fiber: 1g
Sodium: 773mg Protein: 25G

PCLinuxOS Puzzled Partitions

					7		4	
					1			9
7			4		8		1	
1	3				9	2		8
	7							
	8	5				6		1
			2	3	6			
8	2	9						5

SUDOKU RULES: There is only one valid solution to each Sudoku puzzle. The only way the puzzle can be considered solved correctly is when all 81 boxes contain numbers and the other Sudoku rules have been followed.

When you start a game of Sudoku, some blocks will be pre-filled for you. You cannot change these numbers in the course of the game.

Each column must contain all of the numbers 1 through 9 and no two numbers in the same column of a Sudoku puzzle can be the same. Each row must contain all of the numbers 1 through 9 and no two numbers in the same row of a Sudoku puzzle can be the same.

Each block must contain all of the numbers 1 through 9 and no two numbers in the same block of a Sudoku puzzle can be the same.

SCRAPPLER RULES:

1. Follow the rules of Scrabble®. You can view them here. You have seven (7) letter tiles with which to make as long of a word as you possibly can. Words are based on the English language. Non-English language words are NOT allowed.
2. Red letters are scored double points. Green letters are scored triple points.
3. Add up the score of all the letters that you used. Unused letters are not scored. For red or green letters, apply the multiplier when tallying up your score. Next, apply any additional scoring multipliers, such as double or triple word score.
4. An additional 50 points is added for using all seven (7) of your tiles in a set to make your word. You will not necessarily be able to use all seven (7) of the letters in your set to form a "legal" word.
5. In case you are having difficulty seeing the point value on the letter tiles, here is a list of how they are scored:
   - 0 points: 2 blank tiles
   - 1 point: E, A, I, O, N, R, T, L, S, U
   - 2 points: D, G
   - 3 points: B, C, M, P
   - 4 points: F, H, V, W, Y
   - 5 points: K
   - 8 points: J, X
   - 10 points: Q, Z
6. Optionally, a time limit of 60 minutes should apply to the game, averaging to 12 minutes per letter tile set.
7. Have fun! It's only a game!

Download Puzzle Solutions Here

Possible score 219, average score 153.

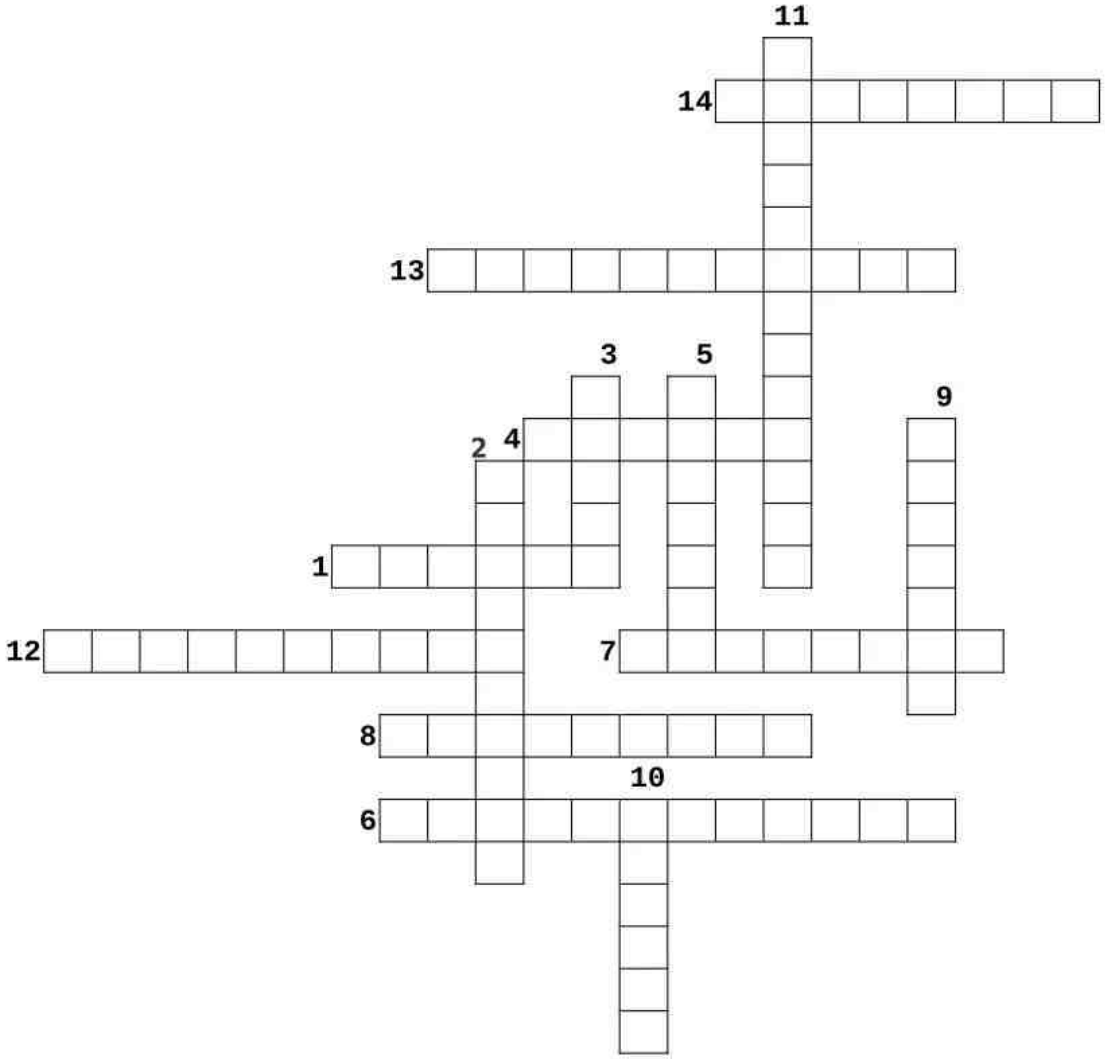
July 2026 Word Find
Summer Food

S L U S H I E S T Y C I P N V Q S P I N A C H S A L A D Y P
G U I J R X U Z S X D E E A T S A P O T S E P B Z M C E S I
A K C X L X G C C R T D B W L D A L A S E S E R P A C E R D
C F R L C J K T Q L E W Y A O B N B A L G L F G Z H L C A T
E X Q Q H X X P X J H L M T U R R N H A R V Y Q R C W F G T
X D T D R F J I P C N I B R U Z E U Z K Q O W V I Y L Z A E
Z O F Q Z Z M Q K E Z J F B F Y L P S K I G S S Q T D C P U
A X X R U V Y D Z V I V S V O G A E S C M P P D H U O U X D
S U H H I L J N A I F K O J B C J J Q R H O Q Q X S O Y I O
T W L U U E L L D C I T J V H M H Q W H P E C R F L K A Y U
A V B X G Q D P A H H N W O Q T M C B T F H T E A C E B L T
M I T J Y P H G I E V E L L I U O T A T A R H T R K I U N E
A B F V M Y J E R K O S N O L E M V D E D P N A A M Q V U B
C G R I L L E D V E G E T A B L E S I F P A B W H U R W X R
I V D J Y Q D Q O Y E I H N V J O Z E W C C W T V Z D O S O
J E X U D H E V H L H N M C R E U X I T A D P U C S A O A S
A A I R G N A S U I Z A T N I M G P N K L K W N Q R L P L Y
R F K H V P X O S M U E W O J V N G E W Q A J O E M A G M R
N N S R R P X K T E V K K A M N E S I E O P T C K W S G O R
O C E T B R G M O P S D U G I A M C P E E W R O X L K F N E
L O I K R N O K U I E X N J H I T I P T B F U C L X A W B B
E E H M C U O X D E V M G J G S A O M O T U F E Q Y E V U P
M S T N E I W P G K J M Q X H W Q N E Z L I R O D E T T R S
R U O W K I H E U V L P I Q D A F V P S D L R G C G S J G A
E B O F N X C C L O T D A L A S A O N I U Q A A E D Q H E R
T F M F F P K Q Q S E A F O O D B O I L Z M Y C M R E F R O
A Y S N C A R V R B Q E A W X B Y H G I T Z J T S I S C S V
W W U V V Y P P Q K B I R T E B R O S T Q N A M V V S S I W
O Q G U D X V P J C J Y F R U I T P O P S R M I H I A U A J
B Z T C E C I D E V A H S Z P A P Q B S E N O C W O N S X X

BBQ CHICKEN	BRUSCHETTA
CANTALOUPE	CAPRESE SALAD
CEVICHE	COCONUT WATER
CRAB CAKES	FRIED GREEN TOMATOES
FRUIT POPS	GAZPACHO
GRILLED VEGETABLES	HAWAIIAN PIZZA
ICED COFFEE	JICAMA
KEY LIME PIE	MELONS
PEACH COBBLER	PESTO PASTA
POPSICLES	QUINOA SALAD
RASPBERRY SORBET	RATATOUILLE
SALMON BURGERS	SANGRIA
SCALLOP CEVICHE	SEAFOOD BOIL
SHAVED ICE	SLUSHIES
SMOOTHIES	SNOW CONES
SORBET	SPINACH SALAD
STEAK SALAD	TIRAMISU
VEGGIE BURGERS	WATERMELON

Download Puzzle Solutions Here

July 2026 Crossword
Summer Food



1. A type of edible seed prepared similar to rice.
2. A type of melon with sweet, aromatic, and usually orange flesh.
3. A dish made typically of flattened bread dough spread with a savory mixture of sauce, cheese & meat.
4. A large, edible, tuberous root of a tropical American plant that looks similar to a potato.
5. An iced punch typically made of red wine, fruit or fruit juice, and soda water.
6. Salad made from tomato, mozzarella, basil, and olive oil.
7. A cold seasoned tomato soup.
8. A ground-up ice dessert commonly served in paper cones, topped with flavored syrup.
9. A traditional Spanish dish prepared with raw fish marinated in lemon juice.
10. A fruity frozen dessert, which is completely dairy-free.
11. A mix of wild rice and brown rice with onion, celery, cashews, pecans and figs.
12. An Italian appetizer consisting of grilled bread topped with garlic, olive oil, and salt.
13. A seasoned stew made of eggplant, tomatoes, green peppers, squash, and sometimes meat.
14. A no-bake Italian dessert featuring layers of coffee-soaked ladyfingers and a rich, creamy mixture of whipped eggs, sugar, and cheese.

Download Puzzle Solutions Here

Mixed-Up-Meme Scrambler

ELLIS

CARPH

GLIJEN

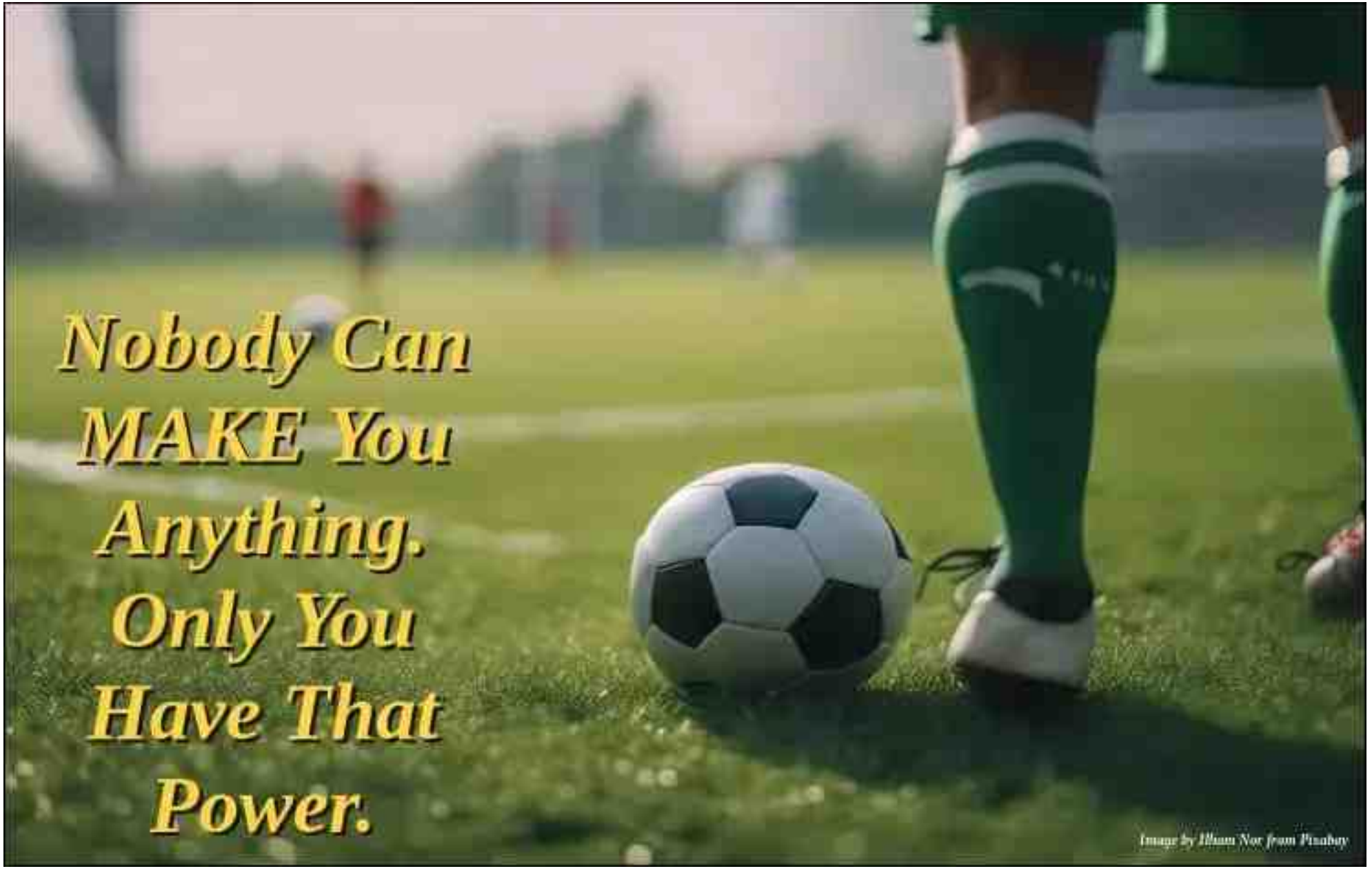
THARRE

What he experienced in
the low part of the ocean ...

A _ _ _ _ "_ _ _ _"

Download Puzzle Solutions Here

Inspiration & Motivation



More Screenshot Showcase

Posted by francescoinblack, on June 13, 2026, running icewm.

Posted by Meemaw, on June 17, 2026, running Xfce.

Posted by mutse, on June 3, 2026, running Mate.

Posted by tbs, on June 18, 2026, running KDE.