The PCLinuxOS magazine

Volume 232

May, 2026

ICYMI: Linux 7.1 Kernel Phasing Out i486 CPU Support

Recent Improvements to DNF Package Manager

Wiki Pick: Disk Space Checking

GIMP Tutorial: Making Custom Brushes

Two New PDF Scripts (With GUI) To Put In Your Toolbox

Tip Top Tips: Using DNF Package Manager With Openbox

Blocking The Internet Archive Won't Stop AI, But It Will Erase The Web's Historical Record

The Government Uses Targeted Advertising To Track Your Location

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:

YouCanTo David Pardue

Alessandro Ebersol

Contributors:

From The Chief Editor's Desk

On April 1, 2026, Artemis II launched from the Kennedy Space Center with four astronauts aboard. Its destination: the moon. Our moon.

For the first time in over 50 years, mankind was headed back to THE FRICKIN' MOON.

With Artemis II, four astronauts were strapped into the Orion spacecraft, perched atop the Space Launch System (SLS). For the first time in over 50 years, a manned spacecraft ventured farther than just low Earth orbit. For some reason, all we've been able to achieve in 50+ years was low Earth orbit.

That's not to say that all of those low Earth orbit missions were a waste. But even after mankind walking on the moon … our moon … spending months on a space station didn't have quite the same luster. At least, not in the sense of manned exploration. Sure, we gained a LOT of information from all of that time spent circling the Earth that will pay untold dividends as we expand to longer and longer manned space missions – both to the moon and beyond.

I've been interested in manned spaceflight since I was a kid. I watched Gemini flights, and then every single Apollo mission. In fact, the entire history of manned spaceflight fits into my lifetime. Back in July, 1969, I was eight years old, less than a month away from my ninth birthday. Apollo 11 didn't just send men to the moon. They actually WALKED on the moon.

I remember talking my parents into allowing me to stay up very late into the night to watch the first images of Neil Armstrong actually setting foot on the moon. It was carried LIVE on all of the television stations. For me, the definitive station to watch was CBS, with Walter Cronkite manning the news desk at such a memorable and significant news event. There was no mistaking or hiding his excitement in that historical moment. I shared his excitement and enthusiasm, as did (I imagine) so many others across the globe.

Fast forward to 2026. We … the entire family … were all dressed and ready for karate class. In fact, we were almost late for karate class. We were glued to the television, streaming NASA TV from YouTube of the Artemis II launch.

I have to admit that, for me, it was a very emotional moment. All of a sudden, all of my memories and emotions from July, 1969 came flooding back to me. I looked over at my two kids, and they were as awed and excited as I was back when I was eight years old. Yes, tears welled up in my eyes. Tears of happiness.

I felt so honored to share this historic return to the moon with my kids. I'm ecstatic that they will get to witness mankind's return to the moon – and possibly beyond – in their lifetimes. For them, I hope that it fuels a lifelong fascination with manned spaceflight in them, just as it did for me all those years ago.

Even though Artemis II didn't land on the moon, and none of the astronauts walked on the moon,

we had finally returned to the moon, even if it were just a "fly by." From the sound of things, Artemis II is the first step towards making grand, manned visits back to the moon. There's even talk of a manned scientific outpost on the moon … eventually. A "lunar base," if you will.

Thus far, only 12 humans have ever set foot on the moon. From the sound of things, that number could climb significantly and rapidly as the Artemis/Orion missions move forward. Meanwhile, Artemis II completed its mission with the safe return of all four astronauts in a bullseye splashdown in the Pacific Ocean on April 10, 2026.

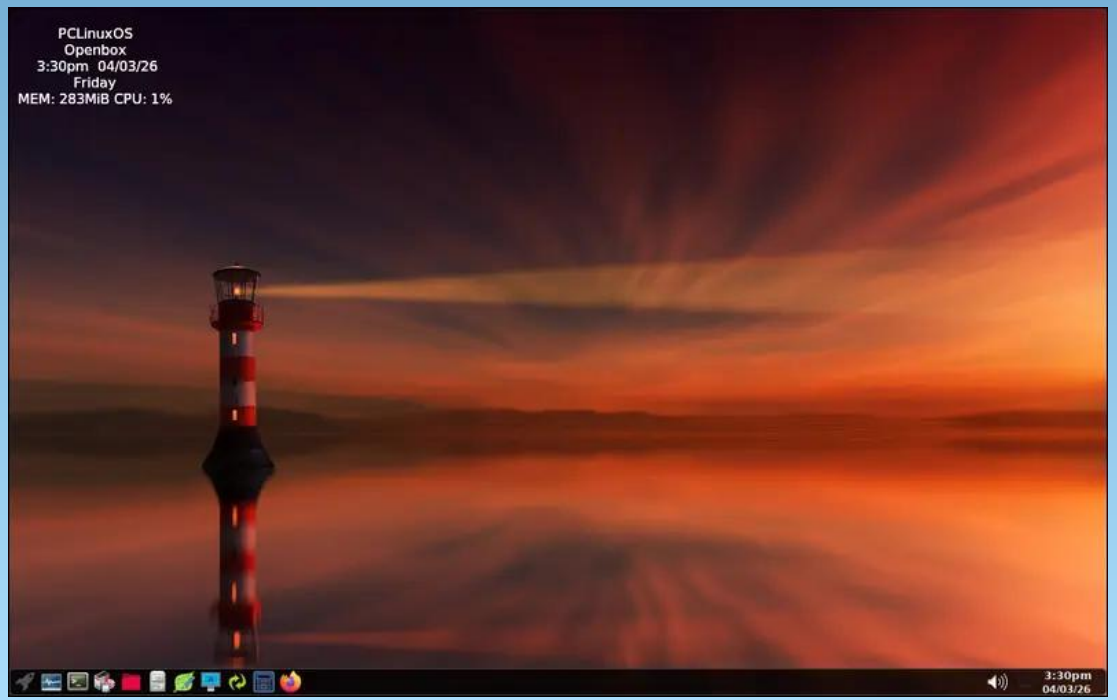
****

This month's cover image is from NASA's Flickr account, and shows the launch of Artemis II. I felt it was fitting after a 50+ year wait to return to the moon. If you haven't kept up with the Artemis II mission, you can get started "catching up" with this article from LiveScience Plus.

****

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

Screenshot Showcase



Posted by astronaut, on April 3, 2026, running openbox.

Two New PDF Scripts (With GUI) To Put In Your Toolbox

by Paul Arnote (parnote)

As you might imagine, I deal with PDF files … like … ALL THE TIME. As such, I have just about every command line tool that I think might be helpful for working with PDFs installed on my computer. But, even with those tools installed, it's easy for me to forget about them, and to reach for a tool that has a GUI.

Why, you might ask? Well, primarily because every time I use one of the command line tools, I'm scrambling around like a chicken trying to get his/her fair share of scratch grains, as I try to remember the command line options for each tool to get what I'm after. After a while, it becomes tiring, which makes reaching for a GUI tool all that more tempting.

That main GUI tool I usually reach for is Master PDF Editor. The best way to think of Master PDF Editor is as a word processor whose default output is PDF files. Plus, it also allows you to edit most PDF files to replace images or text, and then output those changes back to a PDF file. Don't get me wrong … it's a mighty powerful program that excels at its job. But sometimes, it's overkill, and sometimes too slow.

It's not slow in the sense that the program runs slow. But let's set the scenario. Let's say I need to access the images in a PDF file. As in, all of them. I can load the PDF into Master PDF Editor, and save each image in the file, one at a time. That can easily take over 30 minutes to do. In the end, I have the images I need, but I'm left with a feeling that there has to be a faster, less labor-intensive way to grab those images.

Thankfully, there is.

In the April 2026 issue of The PCLinuxOS Magazine, we reprinted a tip from kalwisti in the Tip Top Tips column. Kalwisti posted a "teaser" of his tips in the forum, and linked to a PDF document he created for the full tip. So, to reprint it in the April 2026 issue, we needed to be able to access the text and images in the PDF (his format wasn't compatible with the process we use to lay out the magazine).

So, I loaded the PDF into Master PDF Editor, and proceeded to click on and save each individual image … 16 times. That took about 15–20 minutes to do. And that's when I started thinking "there has to be a faster, easier way" to grab those images.

That's when I remembered the command line PDF tools I have amassed (they are all installed from the PCLinuxOS repository). When I looked at the tools and their command line options, I realized that they were ripe for creating a custom script. Of course, I made sure that the custom script has GUI elements, provided courtesy of Zenity.

I'm sure I could have written an email to kalwisti, and just as certain he would have sent them to me in very short order. But, why bother kalwisti if I didn't have to? That set me off on a journey to create the bash script. Once I had a preliminary working version of my script, I also made a second one to grab all the text elements from a PDF file, as well.

"Easy As PIE"

The first script I came up with extracts all of the images from a PDF file. I call it **PIE**, short for **PDF Image Extractor**.

As with many of the custom bash scripts I write, this one can be run either as a standalone script from the command line, or it can be used as a Thunar Custom Action (just in case you didn't know, I'm a HUGE Xfce fan). You can also type the script in by hand, or you can download the scripts from the magazine server. When you do, save the file(s) to the directory where you store all of your scripts. Then, remove the ".txt" file extension, and make the script executable. You can download PIE from here, if you wish.

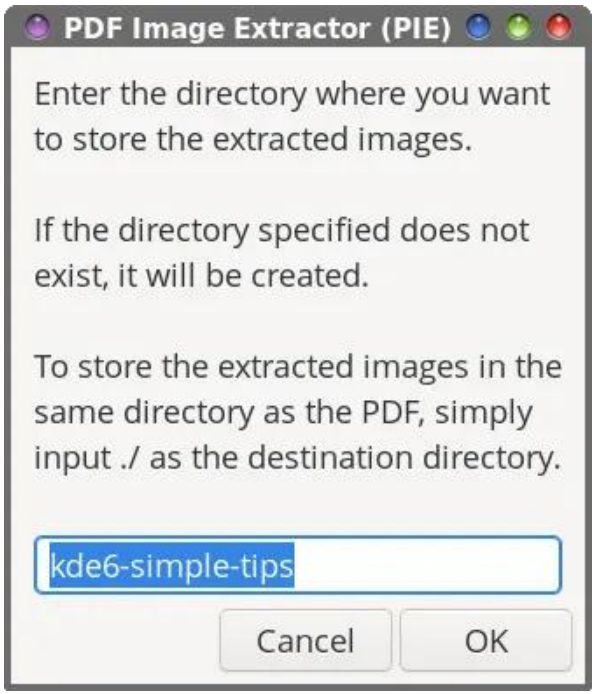
The entire script is 2.0 KiB in file size, so it should download in the literal blink of an eye.

Hopefully, the directory where you store your scripts is also in your $PATH statement. That way, you only need to enter the script name, followed by any command line parameters. If the directory where you store your scripts is NOT in your $PATH statement, you will need to use the full path to the script, the script name, and any command line parameters it takes. As you can see, having your scripts stored in a directory that is also in your $PATH statement can make life a LOT simpler.

Before we begin dissecting the script, let's take a look at the entire script.

```
01. #!/bin/bash
02.
03. # This bash script is free under the GPL v. 2.0 license, and may be
04. # freely distributed in accordance with the terms of said license.
05. # Written by Paul Arnote, Chief Editor of The PCLinuxOS Magazine
06. # for the May 2026 issue of The PCLinuxOS Magazine.
07. #
08. # Usage: PIE.sh <name of PDF file>
09. # May also be used as a Thunar Custom Action, with a custom action
10. # command line of PIE.sh %n. File pattern: *.pdf;*.PDF
11. # Appearance: "Other files"
12. #
13. # This script will extract all of the valid images from a PDF file,
14. # and save them in a subdirectory you choose. If the specified
directory
15. # does not exist, it will be created for you.
16.
17. INPUT=${1,,}
18. DFLT=`basename $INPUT .pdf`
19.
20. DIR=$(zenity --entry --width=300 --height=250 --title="PDF Image
Extractor (PIE)" --text="Enter the directory where you want\nto store the
extracted images.\n\nIf the directory specified does not\nexist, it will be
created.\n\nTo store the extracted images in the\nsame directory as the
PDF, simply\ninput ./ as the destination directory.\n" --entry-text=$DFLT)
21.    if [ $? == 1 ]; then
22.       exit
23.    fi
24.
25. DIR=$DIR
26.    if [ ! -d ./$DIR ]; then
27.       mkdir ./$DIR
28.    fi
29. sleep 1
30.
31. BASE=$(zenity --entry --width=350 --title="PDF Image Extractor (PIE)"
--text="Enter the base name for the\nextracted images.\n\nThe file
extensions will be\nadded automatically, and the\nextracted images will
be\nnumbered sequentially based on\nthe base name you enter below.\n" --
entry-text="PIE-"$DFLT)
32.   if [ $? == 1 ]; then
33.      exit
34.   fi
35.
36. pdfimages -all $1 ./$DIR"/"$BASE | zenity --progress --title "PDF Image
Extractor (PIE)" --width=350 --text="Please wait ...\n\nExtraction In
Progress...\n" --pulsate --auto-close --auto-kill
37.
38. cd ./$DIR
39.
40. if [ $DIR == "./" ]; then
41.    FILE=$(pwd)
42.    else
43.    FILE=$DIR
44. fi
45.
46. COUNT=$(ls -1f | grep $BASE | wc -l)
47.  zenity --info --width=250 --title="PDF Image Extractor (PIE)" --
text=$COUNT" image files extracted from PDF,\nand are saved in $FILE."
48.
49. exit 0
```

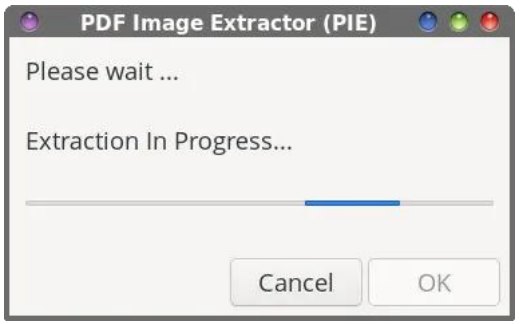
As with all bash scripts, the first line is the "shebang" that gets everything started. Lines 3 through 15 are merely comment lines, outlining (briefly) what the script does, its release terms (GPL v2.0), and some brief usage notes.



In **line 17**, the entire filename of the input file is changed to all lowercase, and then we set a "global" variable (DFLT) (**line 18**) to hold the base filename of the PDF we're extracting images from (which defaults to the same filename as the PDF, but with the PDF file extension removed, and the entire filename changed to lowercase). **Line 20** sets up the first Zenity dialog box. Within that box, we give the end user some brief instructions on what to do, set the width and height of the Zenity dialog box, and set a default subdirectory to store the extracted images in (--entry-text=$DFLT). Of course, the user can override that suggestion just by typing in a different directory name. Unless you know a PDF file doesn't contain a lot of images, storing them in the same directory as the PDF shouldn't be too much of a problem. But if a PDF has a lot of images in it (during testing, I extracted the images from the March 2026 issue of The PCLinuxOS Magazine … all 262 of them; Meemaw extracted images from another PDF magazine, and ended up with over 1,400 images), you'll stay much more organized by extracting the images to a subdirectory of the directory that contains the PDF file you want to extract images from. Hey … we might as well try to stay neat and tidy here!

Of course, you don't ***have*** to accept the default value (that's already filled in for you). You're free to use any directory name you wish. Just keep in mind that whatever you decide to call the directory, it will be set up as a subdirectory of the directory that contains the PDF you're trying to extract images from.

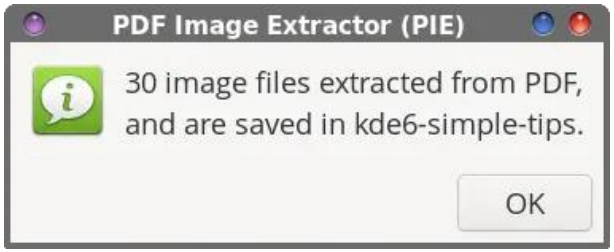
Lines 21 through 23 allow the script to exit "gracefully" if the user hits the "Cancel" button in the dialog box. **Lines 25 through 29** check to see if the desired directory to put the images in already exists, and if it doesn't, it creates the directory. In **line 31**, we invoke our second Zenity dialog box, asking the end user to specify a "base" name for the extracted image files. The text entry value "defaults" to PIE-<default-base-name>. That means, all of your images will start with PIE-<base-name>, with sequential numbers appended to the end of the filename. By using this image naming convention, we're able to keep all of the images grouped together. And, just as with the directory dialog box, you're free to enter whatever base name you want. Accepting the "default" value just makes things easier, in my opinion.



Just as before, **lines 32 through 34** allow the script to exit "gracefully" if the user should click on the "Cancel" button in the Zenity dialog box. The "heavy lifting" for the extraction of the images from the PDF file happens in **line 36**. The **pdfimages** command is the command line tool that's responsible for extracting the images. The **-all** parameter tells pdfimages to extract all of the image formats it recognizes.

With the **./$DIR"/"$BASE** parameter, we're telling pdfimages what directory/subdirectory to save the images to. We then pipe that command to a Zenity progress dialog box with a pulsating progress bar, so we know that the script is still working.

In **line 38**, we issue the **cd** command, just to be sure we keep the script in our current working directory. **Lines 40 through 44** set the formatting for displaying the location where the extracted images are saved to in what I call the "confirmation" dialog box that signals that the task has been completed. In **line 46**, we count how many files have been extracted from the PDF, so we can display it in the confirmation dialog box.

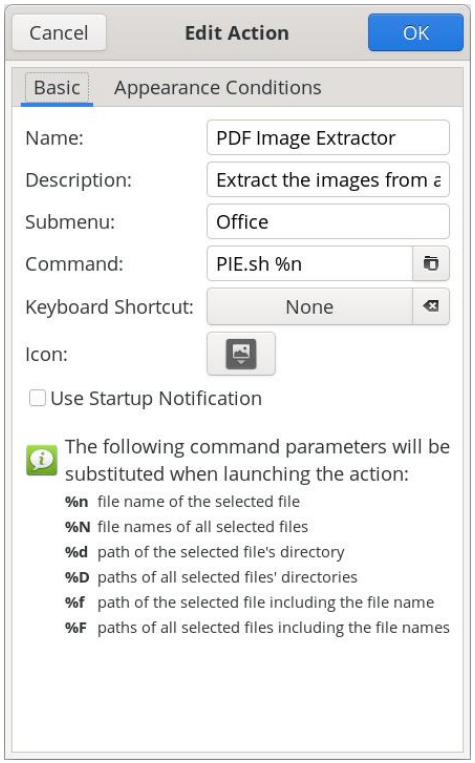


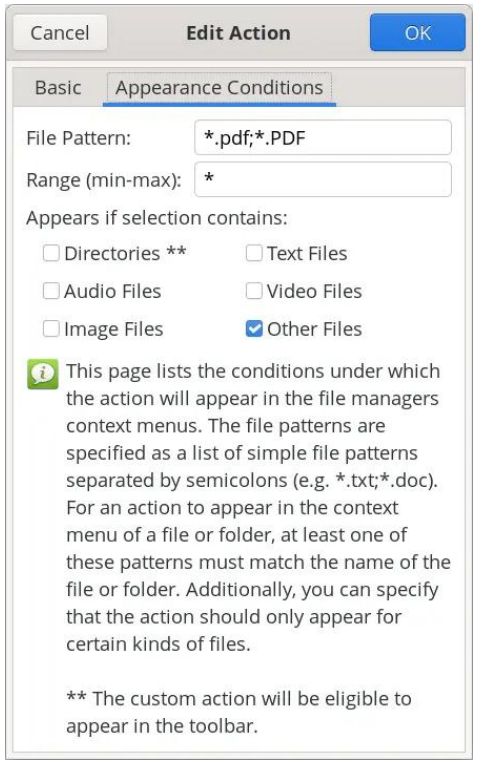
Line 47 sets up a Zenity "Information" dialog box that is displayed when the task is complete. It tells you how many image files were extracted, and where they were stored. The **exit 0** command in **line 49** allows the script to exit gracefully.

Running PIE

The script, as I mentioned earlier, can be run as a standalone script from the command line, or as a Thunar Custom Action. I have no doubt that other file managers can also add the script to their context menus, such as Dolphin and Caja, but that is beyond the scope of this article (and beyond my current skill set, since I only ever run Xfce).

To run it as a standalone script from the command line, the script only takes one command line argument. That is the name of the PDF file you want to extract images from. When running the script from a terminal session, it will help keep you properly oriented by using the cd command to move to the directory that contains the PDF you want to extract images from, and then running it from there. Then, it's just a matter of following the prompts from the Zenity dialog boxes to input the necessary data it needs to complete its task.

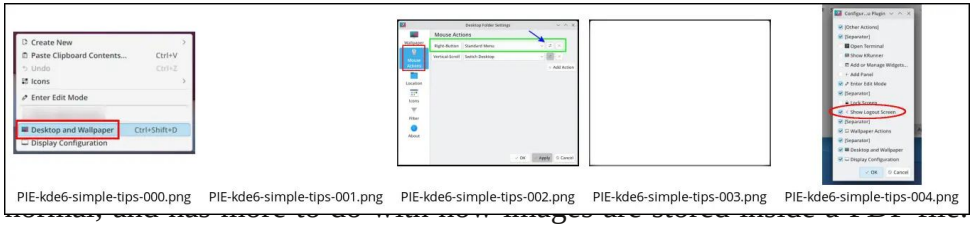




To set up the script as a Thunar Custom Action, make your custom action dialogs resemble those shown above. In the image on the left (showing the "**Basic**" tab), fill in the "Name" field (PDF Image Extractor), the "Description" field (Extract the images from a PDF file), the "Submenu" field (if you use one … I have mine located in an "Office" submenu), and the "Command" field (PIE.sh %n).

In the "**Appearance Conditions**" tab, set the "File Pattern" (*.pdf;*.PDF), leave the "Range" field set to its default value of "*", and place a check in the box next to "Other Files." Once you have those all filled in, save it by clicking on the "OK" button.

Keep in mind that the directory containing my scripts (/home/parnote/bin) IS in my $PATH statement, meaning I can call the script directly, without having to provide a path to the script. Again, if you store your scripts in a directory that IS NOT in your $PATH statement, you will need to provide the full path to the script in the "Command" field.



Now mind you, I'm no "expert" on the internal file structure of PDF files, but the first image is what we would recognize as "the" image, while the "blank" image holds the alpha channel data so that the PDF file can display transparent PNG files. You can keep the blank files around, or delete them. It's up to you.

The **pdfimages** tool recognizes several graphic file formats. Some of them are easy enough to discern … JPEG, PNG, TIFF, JPEG2000 (JP2), JBIG2, CCITT. They are listed if you type "pdfimages --help" at a command line prompt. Support for "other" graphic formats isn't as easy to discern, since they are not listed specifically. For example, here at The PCLinuxOS Magazine, we've been using WEBP graphics for a couple of years now. Even though WEBP graphics aren't listed in pdfimages repertoire, it can "handle" them. What I've been seeing during my "testing" of the PIE script is that pdfimages extracts the "the" image as a JPG file, and then extracts a PNG of the same image to serve as the alpha channel for transparency. Many of the images we use in the production of the magazine have no transparency, but pdfimages has no idea of whether they do or not. So, even if an image has no transparency, pdfimages still extracts "the" image as a JPG file, and a PNG of the same image to serve as the alpha channel image to provide transparency.

I can tell you that AVIF images are "unofficially" supported with pdfimages. None of my "usual" PDF creation tools are capable of using AVIF files, and certainly not Scribus, the tool I use to create many PDFs and the tool we've used monthly to create The PCLinuxOS Magazine for many years. So, to "test" AVIF files compatibility, I converted an image to the AVIF format (using ImageMagick), and then printed it to a PDF file from within GIMP. When I extracted the image from the PDF file, pdfimages wrote the image out to my drive as a perfectly valid PNG file.

Meanwhile, you're going to be quite pleased with how fast PIE works. On my computer, I "timed" the script by running it from the command line, and running the "time" command, followed by the PIE script. On kalwisti's PDF, it extracts 30 images (there are really only 16, but remember the production of additional images for the reasons we just discussed) in just 8.558 seconds. Included in that overall time is 3.291 seconds waiting on me to press the "Enter" key (I just accepted all the defaults), and 0.139 seconds for the system.

Is PIE perfect? Nope. During testing, Meemaw and I found a few PDFs where the images didn't extract well. But, PIE was able to extract the images from the vast majority of PDF files we threw at it. It's hard to know exactly what the production team of those PDF files did differently when they produced the PDF file. It's also hard to know what the production manager or project managers are thinking. Without knowing *exactly* what they did or what they were thinking, most of the "conclusions" we can reach are totally speculation. Did they use a non-standard or unsupported image format? Did they include a custom color palette? Unless we were present when the PDF was created, there's no real way to know exactly what was done, or why.

However, I'm pretty sure you'll discover, just as Meemaw and I did during our testing of the script, that PIE is capable of handling the vast majority of PDF files you might throw at it, and successfully extracting the images from them.

A New PET In The House

The second script I created was born from my work on PIE. Oftentimes, it's not enough to just extract the images from a PDF file. There certainly are times when you will also need to extract the text from a PDF file. Enter **PET**, short for **PDF Extract Text**. PET will extract all of the text elements from a PDF file, and save them to a plain text file.

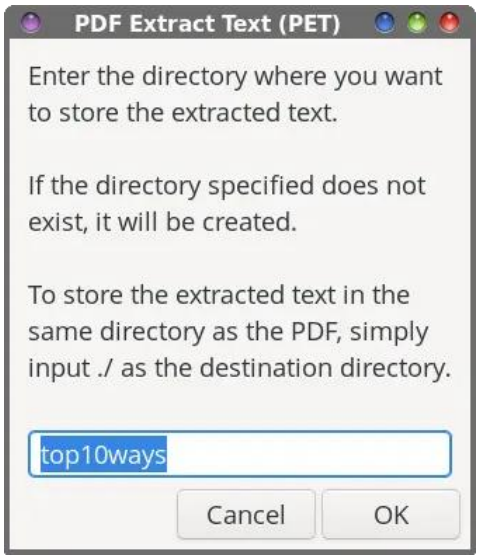
Follow the same instructions for obtaining PET as I gave for PIE. You can download the script from here. The PET script is 1.8 KiB in file size, so it should also download in a literal blink of an eye.

So, before we get into dissecting the script, let's take a look at it.

```
01. #!/bin/bash
02.
03. # This bash script is free under the GPL v. 2.0 license, and may be
04. # freely distributed in accordance with the terms of said license.
05. # Written by Paul Arnote, Chief Editor of The PCLinuxOS Magazine
06. # for the May 2026 issue of The PCLinuxOS Magazine.
07. #
08. # Usage: PET.sh <name of PDF file>
09. # May also be used as a Thunar Custom Action, with a custom action
10. # command line of PET.sh %n. File pattern: *.pdf;*.PDF
11. # Appearance: "Other files"
12. #
13. # This script will extract all of the text elements from a PDF file,
14. # and save them in a subdirectory you choose. If the specified
directory
15. # does not exist, it will be created for you.
16.
17. INPUT=${1,,}
18. DFLT=`basename $INPUT .pdf`
19.
20. DIR=$(zenity --entry --width=300 --height=250 --title="PDF Extract Text
(PET)" --text="Enter the directory where you want\nto store the extracted
text.\n\nIf the directory specified does not\nexist, it will be created.
\n\nTo store the extracted text in the\nsame directory as the PDF,
simply\ninput ./ as the destination directory.\n" --entry-text=$DFLT)
21.   if [ $? == 1 ]; then
22.         exit
23.   fi
24. DIR=$DIR
25. if [ ! -d ./$DIR ]; then
26.    mkdir ./$DIR
27. fi
28. sleep 1
29.
30. BASE=$(zenity --entry --width=350 --title="PDF Extract Text (PET)" --
text="Enter the base name for the\nextracted text.\n\nThe .txt file
extension will be added\nautomatically, and then will be based\non the base
name you enter below.\n\n" --entry-text=$DFLT)
31.   if [ $? == 1 ]; then
32.         exit
33.   fi
34.
35. pdftotext $1 ./$DIR"/"$BASE".txt" | zenity --progress --title "PDF
Extract Text (PET)" --width=350 --text="Please wait ...\n\nExtraction In
Progress...\n" --pulsate --auto-close --auto-kill
36.
37. cd ./$DIR
38.
39. if [ $DIR == "./" ]; then
40.     FILE=$BASE".txt"
41. else
42.     FILE=$DIR"/"$BASE".txt"
43. fi
44.
45. zenity --info --width=250 --title="PDF Extract Text (PET)" --text="Text
extracted from "$1",\nand saved as ./"$FILE
46.
47. exit 0
```

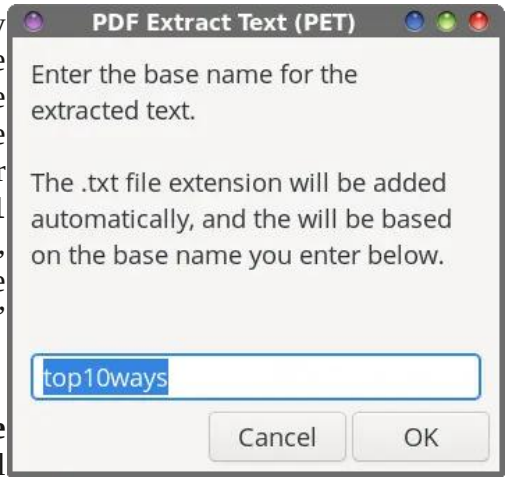
The first 16 lines of the PET script are virtually identical to the first 16 lines of the PIE script, with references to "images" changed to "text." So, I see no need to wade through those lines again.

Line 17 changes the input file filename to all lowercase, to ensure that the basename command in **line 18** works properly. The basename command strips the PDF file extension off of the input file to create our default values in our Zenity dialog boxes.



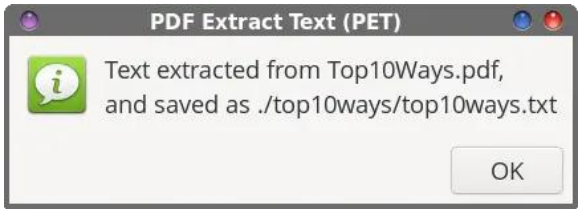
In **line 20**, we set up the first Zenity dialog box. Within that box, we give the end user some brief instructions on what to do, set the width and height of the Zenity dialog box, and set a default subdirectory to store the extracted images in (--entry-text=$DFLT). Of course, the user can override that suggestion just by typing in a different directory name. **Lines 21 through 23** provide a clean way to end the script, if the end user presses the "Cancel" button in the Zenity dialog.

Lines 24 through 28 check to see if the specified directory exists. If it does not, the specified directory is created.



With **line 30**, the second Zenity dialog box is created to allow the user to input the base name for the text extraction. You can accept the default value, or provide another one of your choosing. **Lines 31 through 33** provide that clean, neat way to end the script if the end user chooses the "Cancel" button in the Zenity dialog box.

The "heavy lifting" is done in **line 35**, where the command line tool **pdftotext** extracts the text elements from the PDF. That is piped to a Zenity progress dialog box, with a pulsating progress bar. To be honest, I doubt you'll ever see this. The text extraction is so fast that I've only seen it once or twice, and even then, only for a second or less. Still, I felt it necessary to include it, just in case you ever come across a really large PDF file that takes a bit longer to extract the text elements. This way, you know that the script is running and doing its task, instead of just staring at nothing, unsure if anything is happening.



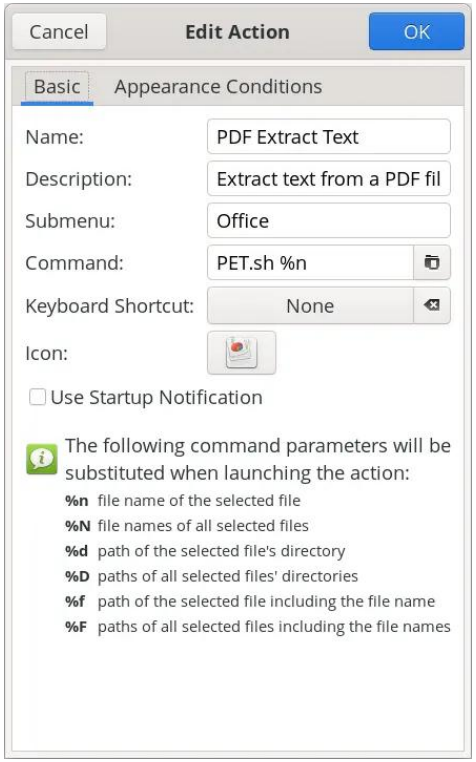
In **line 37**, we issue the **cd** command, just to be sure that the script recognizes the correct working directory. **Lines 39 through 43** set up and format the text that is displayed in the confirmation dialog box when the task is complete. Then, **line 45** displays the Zenity dialog box that lets you know the task has been completed, one I like to call the confirmation dialog box. It informs the end user that the text elements have been extracted from the PDF, and where the resulting ".txt" file has been stored.

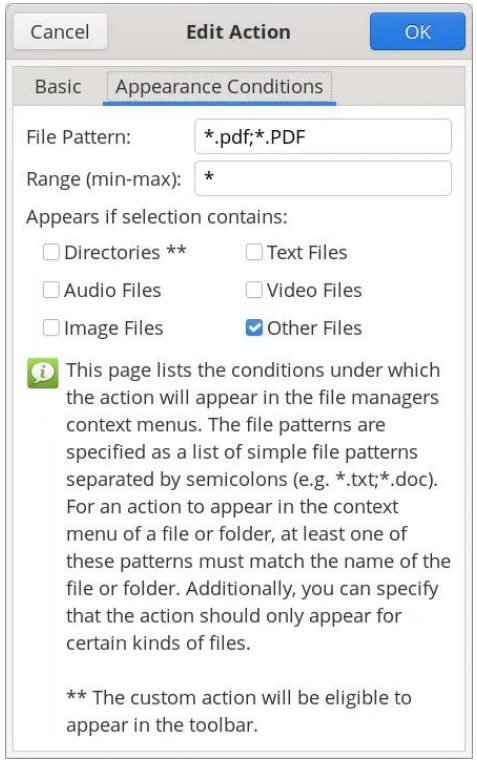
And finally, **line 47** allows us to cleanly exit the script when the task is completed.

Running PET

Like its brethren, PIE, PET takes just one command line option, and that is the filename of the PDF file you want to extract the text elements from. And, just as I did with PIE, PET can be run either as a standalone script from the command line, or as a Thunar Custom Action.

If you're running it from a terminal session, it would be best to cd to the directory that contains the PDF you're wanting to extract the text elements from. It will help to keep things a LOT tidier, and will take away the onerous use of full path statements as the command line parameter.





To set up the script as a Thunar Custom Action, make your custom action dialogs resemble those shown above. In the image on the left (showing the "**Basic**" tab), fill in the "Name" field (PDF Extract Text), the "Description" field (Extract the text from a PDF file), the "Submenu" field (if you use one … I have mine located in an "Office" submenu, with my other custom PDF tools), and the "Command" field (PET.sh %n).

In the "**Appearance Conditions**" tab, set the "File Pattern" (*.pdf;*.PDF), leave the "Range" field set to its default value of "*", and place a check in the box next to "Other Files." Once you have those all filled in, save it by clicking on the "OK" button.

Keep in mind that the directory containing my scripts (/home/parnote/bin) IS in my $PATH statement, meaning I can call the script directly, without having to provide a path to the script. Again, if you store your scripts in a directory that IS NOT in your $PATH statement, you will need to provide the full path to the script in the "Command" field.

```
 1 Financial security doesn't just happen. It takes
 2 planning and commitment and, yes, money.
 3
 4 FACT
 5
 6 Today, only 43 percent of Americans
 7 have calculated how much they need to
 8 save for retirement.
 9
10 FACT
11
12 In 2005, of those who had 401(k)
13 coverage available, 25 percent didn't
14 participate.
15
16 FACT
17
18 The average American spends 20 years
19 in retirement.
20
21 To find out more, call the Employee Benefits
22 Security Administration at 1-866-444-EBSA (3272)
23 and request the following brochures:
24 Savings Fitness:
25 A Guide to Your Money and Your Financial Future
26 Taking the Mystery Out of Retirement Planning
27 What You Should Know about Your Retirement Plan
28 Filing a Claim for Your Retirement Benefits
29 Choosing a Retirement Solution for Your Small Business
30 Women and Retirement Savings
31 Or view them on the Web at: www.dol.gov/ebsa
32
33 The following Web sites can also be helpful:
```

The image above shows sample text that was extracted from a PDF file that Meemaw shared with me when we were testing these scripts (the text file is opened up in Mousepad with line numbering turned on). As you can see, it really is just plain text. To "reuse" the text (provided you have the ***rights*** to reuse the text … don't fail to consider the ramifications and consequences of copyright infringement!), it's going to require some formatting work in whichever tool you're using to reproduce it. But, at least you have the text, so the reformatting should just be something you have to do when you lay it out.

Did I mention that PET is *F-A-S-T*?! It's even faster than PIE at performing its task.

Summary

So, there you have two powerful tools to add to your toolbox that, when used, has the potential to save you a LOT of time. These tools are fast at what they do, and follow the time-honored Linux practice of "do one thing, and do it well."

I can think of several use-case scenarios where these scripts will be useful. Plus, they'll save you considerable amounts of time. Let's say you need to make revisions to a document you created three years ago. You want/need to update that document. While you cannot find the original, you do manage to locate a PDF of the document. With these tools, you can extract the images and text from that PDF, and then redo it … with your changes, of course.

I've already found a use for these scripts, in taking PDF files and extracting the images and text from them. In fact, I can think of several instances where I wish I had this in my arsenal of tools. Every once in a while, we (those of us here at the magazine) have a need to access images in a PDF file. Previously, we'd have to display it in a PDF Viewer, and take a screenshot of the image … and hope for the best. These tools would have made it so, so, so much easier.

PCLinuxOS Recipe Corner

Grandma's Hamburger Casserole

Serves: 6

INGREDIENTS:

1 pound ground beef
2 (15 ounce) cans tomato sauce
1 teaspoon white sugar, or to taste
1 teaspoon garlic powder
Salt to taste
1 (8 ounce) package egg noodles
1 cup sour cream
3 ounces cream cheese, softened
1 large white onion, finely diced
½ cup shredded sharp Cheddar cheese, or more to taste

DIRECTIONS:

Heat a large skillet over medium-high heat. Cook and stir ground beef in the hot skillet until browned and crumbly, 5 to 7 minutes; drain and discard grease.

Mix tomato sauce, sugar, garlic powder, and salt into the ground beef; simmer until flavors blend, about 20 minutes. Remove from the heat and cover the skillet.

While the sauce is cooking, bring a large pot of lightly salted water to a boil. Cook egg noodles in the boiling water, stirring occasionally until cooked through but firm to the bite, 7 to 9 minutes. Drain.

While the egg noodles are cooking, preheat the oven to 350 degrees F (175 degrees C). Grease a 9x13-inch casserole dish.

Mix sour cream, cream cheese, and onion together in a bowl.

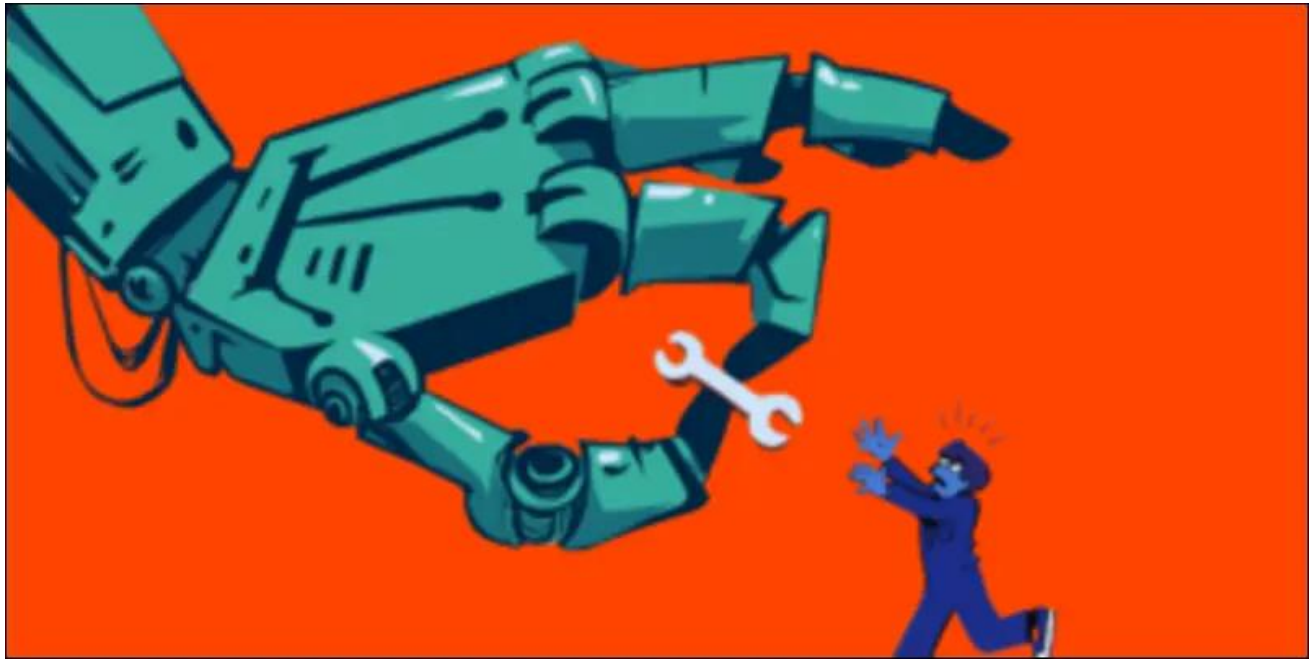
Layer 1/2 of the egg noodles into the prepared casserole dish. Top with 1/2 of the sour cream mixture, then 1/2 of the ground beef mixture. Repeat layers once more. Sprinkle Cheddar cheese over top.

Bake in the preheated oven until the cheese is melted and golden brown, 25 to 30 minutes.

NUTRITION:

Calories: 507 Carbs: 23g Sodium: 975mg
Fiber: 3g Protein: 31g

Print Blocking Is Anti-Consumer: Permission To Print, Part 1



by Rory Mir
Electronic Frontier Foundation

When legislators give companies an excuse to write untouchable code, it's a disaster for everyone. This time, 3D printers are being targeted across a growing number of states. Even if you've never used one, you've benefited from the open commons these devices have created — which is now under threat.

This isn't the first time we've gone to bat for 3D printing. These devices come in many forms and can construct nearly any shape with a variety of materials. This has made them absolutely crucial for anything from life-saving medical equipment, to little Iron Man helmets for cats, to everyday repairs. For decades these devices have been a proven engine for innovation, while democratizing a sliver of manufacturing for hobbyists, artists, and researchers around the world.

For us all to continue benefiting from this grassroots creativity, we need to guard against the type of corporate centralization that has undermined so much of the promise of the digital era. Unfortunately some state legislators are looking to repeat old mistakes by demanding printer vendors install an enshittification switch.

In the U.S., three states have recently proposed that commercial 3D-printer manufacturers must ensure their printers only work with their software, and are responsible for checking each print for forbidden shapes — for now, any shape vendors consider too gun-like. The 2D equivalent of these "print-blocking" algorithms would be demanding HP prevent you from printing any harmful messages or recipes. Worse still, some bills can introduce criminal penalties for anyone who bypasses this censorware, or for anyone simply reselling their old printer without these restrictions.

If this sounds like Digital Rights Management (DRM) to you, you've been paying attention. This is exactly the sort of regulation that creates a headache and privacy risk for law-abiding users, is a gift for would-be monopolists, and can be totally bypassed by the lawbreakers actually being targeted by the proposals.

Ghosting Innovation

"Print blocking" is currently coming for an unpopular target: ghost guns. These are privately made firearms (PMFs) that are typically harder to trace and can bypass other gun regulations. Contrary to what the proposed regulations suggest, these guns are often not printed at home, but purchased online as mass-produced build-it-yourself kits and accessories.

Scaling production with consumer 3D printers is expensive, error-prone, and relatively slow. Successfully making a working firearm with just a printer still requires some technical know-how, even as 3D printers improve beyond some of these limitations. That said, many have concerns about unlicensed firearm production and sales. Which is exactly why these practices are already illegal in many states, including all of the states proposing print blocking.

Mandating algorithmic print-blocking software on 3D printers and CNC machines is just wishful thinking. People illegally printing ghost guns and accessories today will have no qualms with undetectably breaking another law to bypass censoring algorithms. That's if they even need to — the cat and mouse game of detecting gun-like prints might be doomed from the start, as we dive into in this companion post.

Meanwhile, the overwhelming majority of 3D-printer users do not print guns. Punishing innovators, researchers, and hobbyists because of a handful of outlaws is bad enough, but this proposal does it by also subjecting everyone to the anticompetitive and anticonsumer whims of device manufacturers.

Can't make the DRM thing work

We've been railing against Digital Rights Management (DRM) since the DMCA made it a federal crime to bypass code restricting your use of copyrighted content. The DRM distinction has since been weaponized by manufacturers to gain greater leverage over their customers and enforce anti-competitive practices.

The same enshittification playbook applies to algorithmic print blockers.

Restricting devices to manufacturer-provided software is an old tactic from the DRM playbook, and is one that puts you in a precarious spot where you need to bend to the whims of the manufacturer. Only Windows 11 supported? You need a new PC. Tools are cloud-based? You need a solid connection. The company shutters? You now own an expensive paperweight — which used to *make* paperweights.

It also means useful open source alternatives which fit your needs better than the main vendor's tools are off the table. The 3D-printer community got a taste of this recently, as manufacturer Bambu Labs pushed out restrictive firmware updates complicating the use of open source software like OrcaSlicer. The community blowback forced some accommodations for these alternatives to remain viable. Under the worst of these laws, such accommodations, and other workarounds, would be outlawed with criminal penalties.

People are right to be worried about vendor lock-in, beyond needing the right tool for the job. Making you reliant on their service allows companies to gradually sour the deal. Sometimes this happens visibly, with rising subscription fees, new paywalls, or planned obsolescence. It can also be more covert, like collecting and selling more of your data, or cutting costs by neglecting security and bug fixes.

With expensive hardware on the line, they can get away with anything that won't make you pay through the nose to switch brands.

Indirectly, this sort of print-blocking mandate is a gift to incumbent businesses making these printers. It raises the upfront and ongoing costs associated with smaller companies selling a 3D printer, including those producing new or specialized machines. The result is fewer and more generic options from a shrinking number of major incumbents for any customer not interested in building their own 3D printer.

Reaching the Melting Point

It's already clear these bills will be bad for anyone who currently uses a 3D printer, and having alternative software criminalized is particularly devastating for open source contributors. These impacts to manufacturers and consumers culminate into a major blow to the entire ecosystem of innovation we have benefited from for decades.

But this is just the beginning.

Once the infrastructure for print blocking is in place, it can be broadened. This isn't a block of a very specific and static design, like how some copiers block reproductions of currency. Banning a category of design based on its function is a moving target, requiring a constantly expanding blacklist. Nothing in this

legislation restricts those updates to firearm-related designs. Rather, if we let proposals like this pass, we open the door to the database of forbidden shapes for other powerful interests.

Intellectual property is a clear expansion risk. This could look like Nintendo blocking a Pikachu toy, John Deere blocking a replacement part, or even patent trolls forcing the hand of hardware companies. Repressive regimes, here or abroad, could likewise block the printing of "extreme" and “obscene” symbols, or tools of resistance like popular anti-ICE community whistles.

Finally, even the most sympathetic targets of algorithmic censorship will result in false positives—blocking 3D-printer users’ lawful expression. This is something proven again and again in online moderation. Whether by mistake or by design, a platform that has you locked in has little incentive to offer remedies to this censorship. And these new incentives for companies to surveil each print can also impose a substantial chilling effect on what the user chooses to create.

While 3D printers aren’t in most households, this form of regulation would set a dangerous precedent. Government mandating on-device censors which are maintained by corporate algorithms is bad. It won’t work. It consolidates corporate power. It criminalizes and blocks the grassroots innovation and empowerment which has defined the 3D-printer community. We need to roundly reject these onerous restraints on creation.

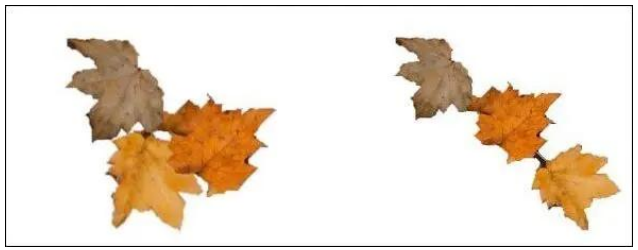
Screenshot Showcase

Posted by bliss, on April 5, 2026, running KDE.

GIMP Tutorial: Making Custom Brushes

by Meemaw

On one of the magazine covers last fall, I had an autumn scene where the leaves were on the ground, and there was a sidewalk in the picture. I made a couple of custom brushes for that. (No matter how much someone sweeps or rakes, there are always still leaves on the sidewalk!)



If you don't know how to make a custom brush for your GIMP, we should do it, so it will help you in one of your future projects.

I searched for tutorials on how to do brushes. Davies Media Design had a tutorial on how to do them. **HOWEVER**, his tutorial used GIMP version 2.10, and GIMP *has changed the method* for version 3.2. GIMP does update their manual when there are changes, so I was able to refer to the manual as well.

There are a couple of ways to do custom brushes. Most of the brushes you already have in GIMP are designed to be able to paint in whatever foreground color you have. In addition, you can make brushes that will stay the same, like the leaves I used.

Let's get started! There are 4 kinds of brushes:

Ordinary - You can create it in the GIMP window. This is probably the way you will create most of them.

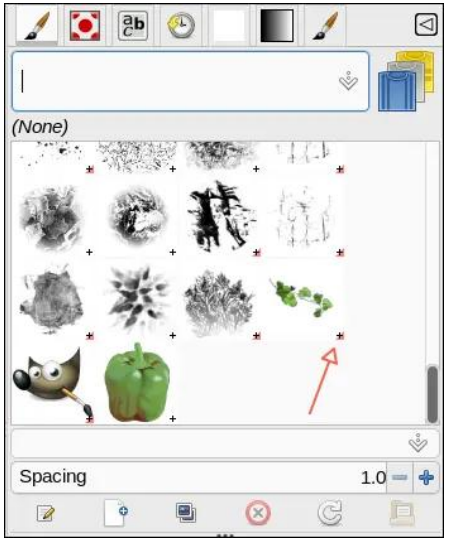
Parametric - Created in the Brush Editor. It is also an ordinary brush but its creation and export is faster from the Brush Editor. The downside is that the variety is limited.

The above two can take on your desired foreground color.

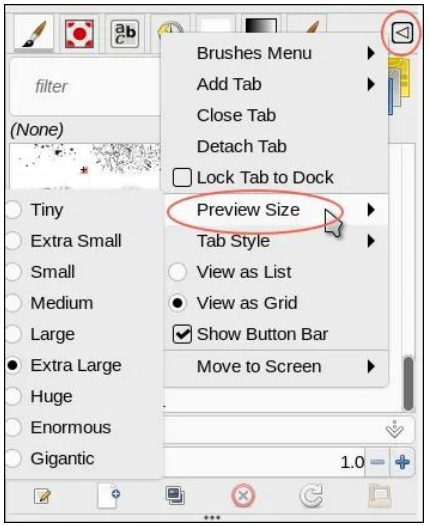
Color - Uses an image to serve as a brush. In some instances (like my leaves), you can use the brush to put them anywhere you want.

Animated - It uses layers to make a 'variable' brush, so when you click, it cycles through the brushes/layers available.

Just in case you didn't know, you will see tiny icons in the corner of a brush. A red triangle in the corner of the brush means it's animated, and if you click and hold, it will cycle through the available brushes. If there is an "+", it means that the brush is actually larger than the icon you're seeing. In the image below, the "vines" brush has a red triangle and a "+". That brush is animated with 3 different designs.



Also, you can increase the preview size of your brushes by right-clicking the triangle in the upper right of that window and choosing "Preview size". You'll see nine choices. I've changed mine, but I believe the default is Large.

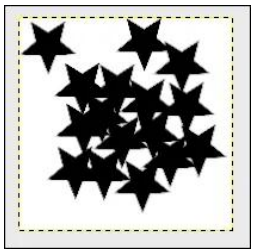


Creating an ordinary brush

The easiest thing you can do is simply copy your creation to the clipboard (<CTRL> + C). GIMP puts your copy into the brushes dialog for your use while you are working on your creation. **NOTE**: The catch here is that your brush is *only* in GIMP until you close it. If you want to use it again after GIMP has been closed and reopened, you have to export it as a brush.

To make a brush, choose **File > New**. In that window, put the size you want. I used 180 x 180 px (the maximum size is 1024 x 1024). Now, under **Advanced Options**, change **Color Space** to **Grayscale**, and **Fill with white**. Mine says Background color, because my background color is white. You can also change it to white in the window.

With your new file open, create the brush you want. I used the star brush already in GIMP to make a group, but your brush could be lots more intricate (center, top).

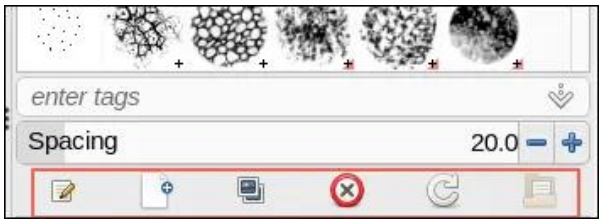


If you want to edit this later, you should save it to GIMP's native .xcf format so you can open it again. To export it as a brush, choose **File > Export as…** Choose a name for your brush, and save it with the .gbr extension. You will get another window, asking for a description and the spacing between brush strokes. If you're holding the mouse button down, the brush will continue to draw. The distance is how far apart the brush strokes are. The description will always say "GIMP Brush" in the brushes dialog unless you change it, so if you want to put your brush name in this window, it will help you distinguish them in the brushes dialog. I have the brushes dialog open in GIMP as well as tool options, because the brushes dialog has some other tools.

Now before you can use these brushes, they have to be in the **GIMP brush directory** rather than one of your folders. In Linux, it should be Home/(YourUserName)/.config/GIMP/3.2/ brushes. (We have version 3.2 now.)

Creating parametric brushes in the Brush editor

The brush editor is very useful, but limited in what you can design, which is why you'll most likely design your brushes directly in GIMP. To use the editor, open your brushes dialog (separate from the brush toolbox). At the bottom of the brushes dialog you'll see some tools. From left to right, Edit brush, New brush, Duplicate brush, Delete brush, Refresh brushes and Open brush as image.



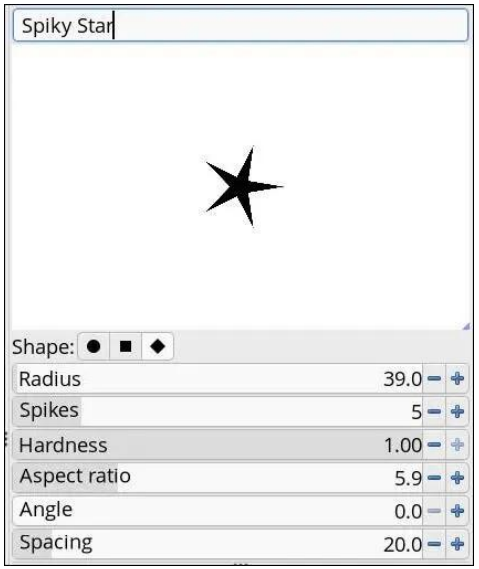
To create a new brush, simply click New brush. The Brush editor will open with a single dot in the center. From there you can edit that single circle according to shape, size, hardness, angle and a couple of other aspects. You can also give it a name. As you can see below, you can choose a circle, square or diamond to work with. I chose a diamond.

Radius decides how big the brush is.
Spikes decides how many sides it has. I chose 5. It made a star like the one that's already in the brushes dialog, but I wanted something different.
Hardness decides whether it's sharp or faded around the edges. 100 means the whole brush is black and there's no blur around the edges.
Aspect ratio is the one that made my star

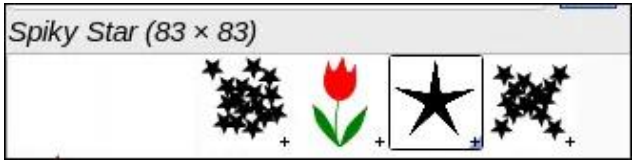
thinner. You can play with it to get the result you want.

Angle is which direction you want it. For the star, a different angle might have one of the points sticking straight up.

Spacing is how far apart you want the brush to draw if you hold your mouse down. A higher number puts each stroke farther away and a smaller number clumps them together more.



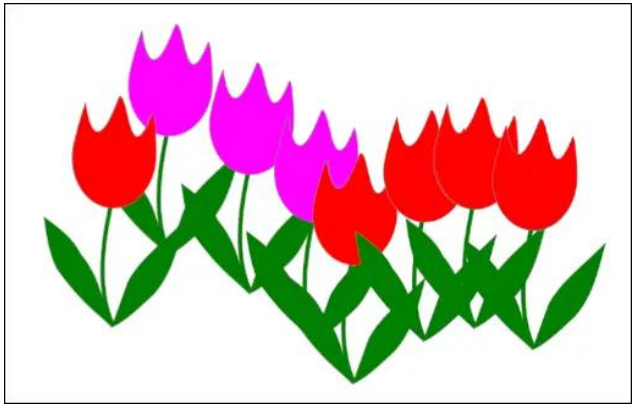
When you get it the way you want it, all you have to do is close the Brushes editor tab in GIMP. The program will save a file in your designated brushes folder. This one will be a text file with a .vbr extension. Mine said Untitled.vbr, but opening the text file has the settings I chose, including the name Spiky Star, so I changed the file name to Spiky Star.vbr. When I open GIMP tomorrow, it will be in my brushes dialog. However, I can always go to the brushes dialog and click **Refresh brushes**. And there it is…



Creating color brushes

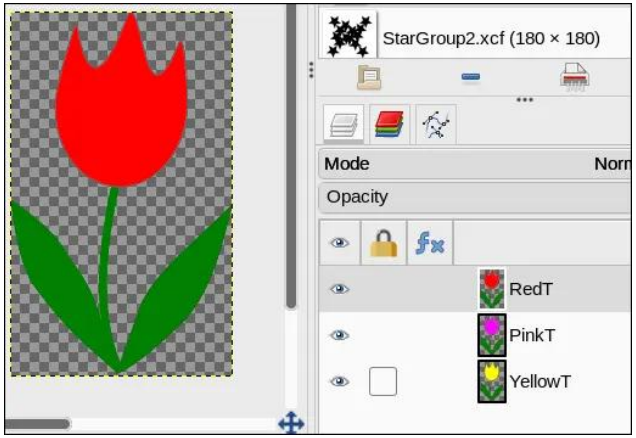
Creating color brushes is a little bit different, because you want to keep the color of the image you're using for the brush. This time, you'll load your image into a new file, but filled with transparency. I had created a simple tulip graphic in Inkscape for another project, and saved it. Now I'll open it in GIMP, along with another file the same size, filled with transparency. This one's 175 x 287 px, so I'll open a new file that is the same size.

You'll want to copy the colored image to the new transparent file. Remember, in the new version of GIMP, the paste will be on a different layer, so you should merge the layers. Now, all that is left is to export it and put it in the GIMP brushes folder, and refresh your GIMP brushes. I've made a red one and a pink one and just clicked in the window.



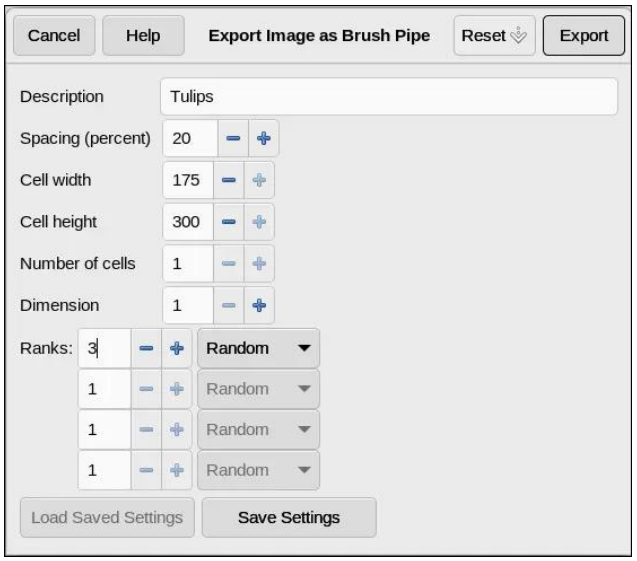
Creating animated brushes

When I created the single brushes, I saved each of them as a GIMP .xcf file, just in case I wanted to edit them later. This has been a very good practice for me since I use GIMP a lot. I opened one of them to start an animated brush. For this, you need to make a separate layer for each brush you want to include. I not only had a red and a pink tulip, I also had a yellow one, so I'll have 3 layers. I copied each color tulip to a transparent layer.



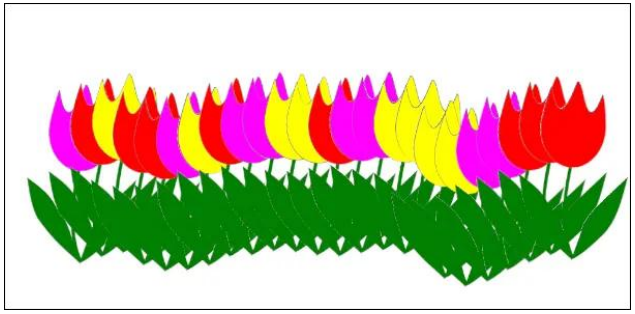
Each tulip is exactly the same as the others, except for the color. Since the red one is on the top layer, it is visible while the others are only visible in the layers dialog.

Again, get your layers the way you want them, then export as a brush. This time, however, the extension is .gih. Also, this time, your export window will be larger (next page, top left).



The settings are more complex:

Spacing is still how far apart you want your strikes when you hold the mouse button down.
Cell width and **cell height** should be the dimensions of your brush.
Number of cells tells how many layers you have.
Dimension tells GIMP to cut that brush into pieces. With the right pattern, you could choose 4, cutting each layer into 4 pieces, and have even more brushes. For our brush, however, we want to see the whole tulip.
Ranks tells GIMP how to order the layers when the brush is used, so for all 3 layers, the setting is Random. When you hold down the mouse button, it will paint them randomly (center, top).



For some reason it says Height Mismatch! I thought that was if your image is on a different-sized background, but all my layers are the same size as the tulips I used. I'll need to research that. Make sure you send your brush to the brushes folder, and refresh your brushes.

I hope you have fun creating any brush you might want to use.

Screenshot Showcase

Posted by francescoinblack, on April 8, 2026, running icewm.

Recent Improvements to DNF Package Manager

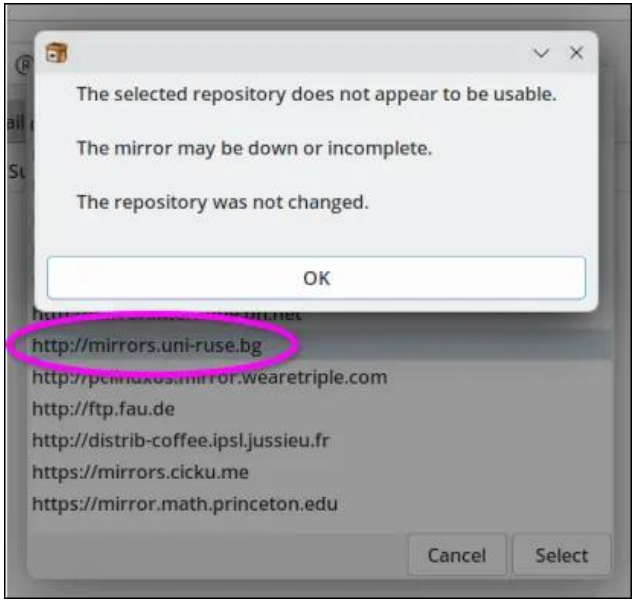
by David Pardue (kalwisti)

DNF Package Manager (the GUI front end to DNF5) was introduced in PCLinuxOS approximately nine months ago. I wrote an overview of this homegrown application for the October 2025 issue of our magazine; it covers basic usage as well as background information on the reasons why the PCLinuxOS development team switched to DNF. In this month's article, I will discuss improvements to DNF Package Manager since its initial release. The current version is 1.14-3, which was released in mid-April 2026.

Ability to Refresh If Repository Is Unavailable

Version 1.14-2 of the DNF Package Manager was released in mid-February 2026. It included a new feature: the program will not freeze up if a repository is unavailable. If a repository is down when selecting a new one, the **Available** tab will display a notification. Thanks to this change, the GUI is still able to refresh and choose a repository that is working without the freeze (center, top).

The screenshot above shows the notification I received when attempting to change my default repository to the Bulgarian PCLinuxOS mirror



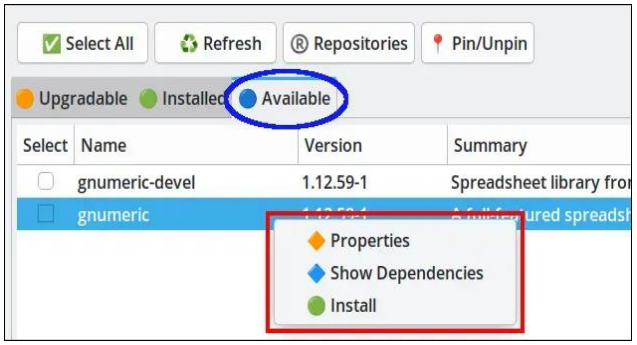
("Angel Kanchev" University of Ruse). It was temporarily unavailable.

Resizable Install and Remove Window

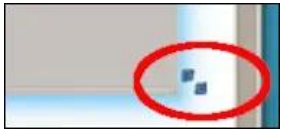
In version 1.14-3, the dialog window height was increased for easier reading and the dialog windows can now be resized if necessary. The screenshots at right illustrate this new enhancement.

With Available/Installable packages

Highlight your desired package (*Gnumeric* in this example), then right-click. A small dialog with several options will appear:



If you click on the **Properties** option, a new window will open. You can resize the window by using the handles at the edge of the window's border:

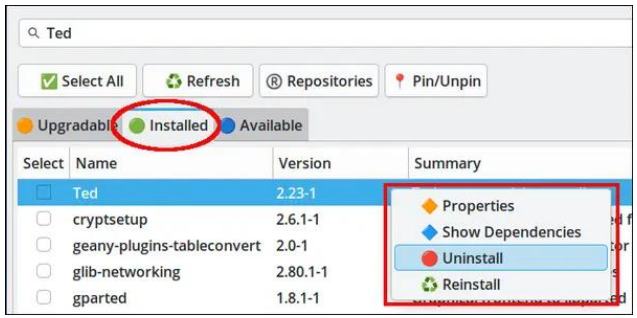


The screenshot above shows the **Properties** window after I resized it.

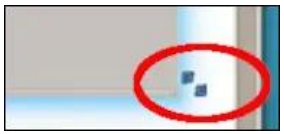
Selecting the **Show Dependencies** option or the **Install** option will also open a resizable window.

Removing Installed packages

Highlight the package that you wish to remove (*Ted* in this example), then right-click. A small dialog with several options will appear:



If you want to inspect Ted's dependencies before uninstalling the text processor, click on the Show Dependencies option. A new window will open, which you can resize by using the handles at the edge of the window's border:



The screenshot at right illustrates the **Show Dependencies** window after being resized.

Selecting the **Properties** option or the **Uninstall** option will also open a resizable window.

Fix dnf_launcher Problem on Non-English Systems

Some PCLinuxOS users reported issues with removing packages via the DNF Package Manager GUI on non-English-language systems. Upgreyed fixed how the **/usr/bin/dnf_launcher** works in non-English locales when launched from the .desktop file, thereby allowing the GUI to work properly with languages other than English.

Remove Older Kernel(s)

This is not a new feature, but I will mention it because it has confused some users. If you need to remove an older/outdated kernel from your system, DNF Package Manager requires you to remove the **kernel-devel** package first. (This differs from the equivalent procedure with Synaptic Package Manager.)

- Press the **Refresh** button to update the package list.

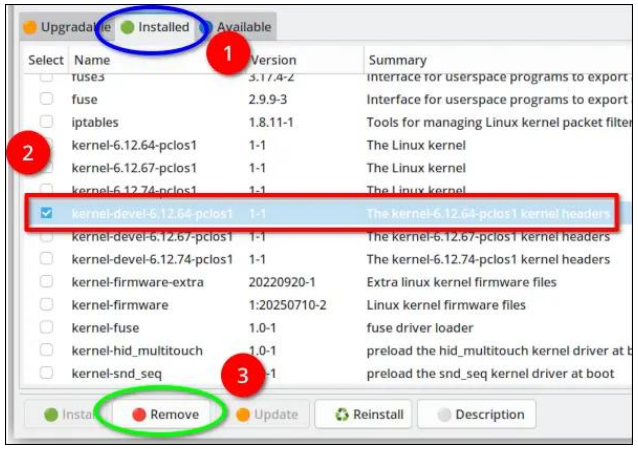
- Click on the **Installed [Packages]** tab to bring it forward. In the Search bar, type "**kernel**" (without the quotation marks) to search for the kernels installed on your system.

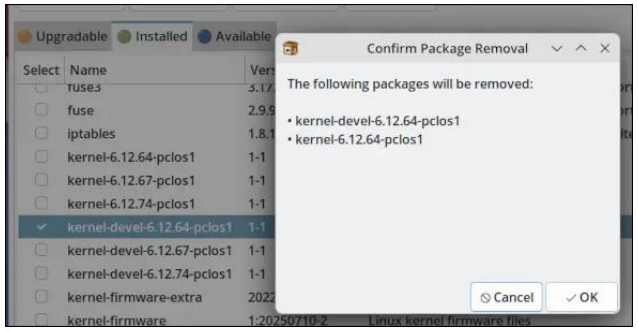
- Scroll down the list of packages retrieved by the search, and find the kernel you wish to remove.

- Select the **kernel-devel** package associated with the kernel you wish to remove, by checking/ticking the box in the **Select** column (to the left of the package name).

- Click on the **Remove** button.

Note: The corresponding **kernel** package will be automatically removed along with the **kernel-devel** package that you selected for removal.





Tip: If you have multiple kernel versions installed, it is generally considered good practice to retain two known working kernels. That way, if a newly installed kernel is not compatible with your hardware and will not boot, or is somehow problematic, you can fall back to the previous kernel. For more information on how PCLinuxOS handles kernels, please consult the PCLinuxOS KnowledgeBase.

Conclusion

I am grateful to Texstar and Upgreyed for continuing to improve DNF Package Manager; they have added features which they believe will be useful. Upgreyed mentioned that more add-ons could be incorporated at a later date — with Texstar's approval, of course. I have used DNF Package Manager exclusively for the past eight months; it has performed reliably, and I have confidence in it.

Some recent forum posts by Texstar plainly state that PCLinuxOS's version of apt-rpm is obsolete, and that Synaptic is obsolete, as it is no longer maintained. Although he has not yet announced an official "sunset" date for Synaptic — and Synaptic is still working at the moment — it is unknown whether Synaptic will continue to work indefinitely. In my opinion, users who have not yet switched to DNF Package Manager would be prudent to begin familiarizing themselves with the application, before Synaptic is retired.

PCLinuxOS favors GUI-based tools because it has always focused on ease of use. However, you may use DNF5 via the command line, if you wish. If you would like to explore some essential DNF5 commands, you can download my PDF guide (see Section 14, p. 15–22). Hunter recently posted a succinct explanation of the difference between "dnf upgrade" and "dnf distro-sync".

Wiki Pick: Disk Space Checking

Relevant to all versions of PCLinuxOS

The easiest way to check your system's disk space is by using the **df** (disk free) command in a terminal window.

The df utility shall write the amount of available space and file slots for file systems on which the invoking user has appropriate read access. File systems shall be specified by the file operands; when none are specified, information shall be written for all file systems.

Syntax: df — report free disk space.

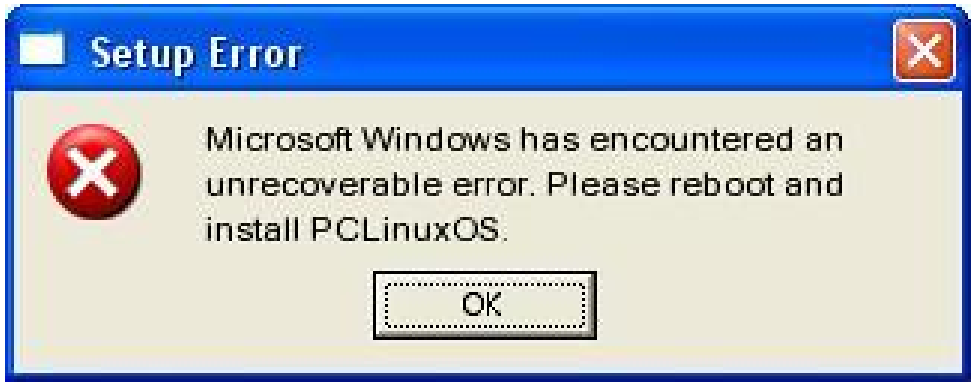
Options: The following arguments shall be supported:

-a include pseudo, duplicate, inaccessible file systems

-h print sizes in power of 1024 (e.g., 1023M)

-k Use 1024-byte units, instead of the default 512-byte units, when writing space figures.

-P Produce output in the format described in the STDOUT section.

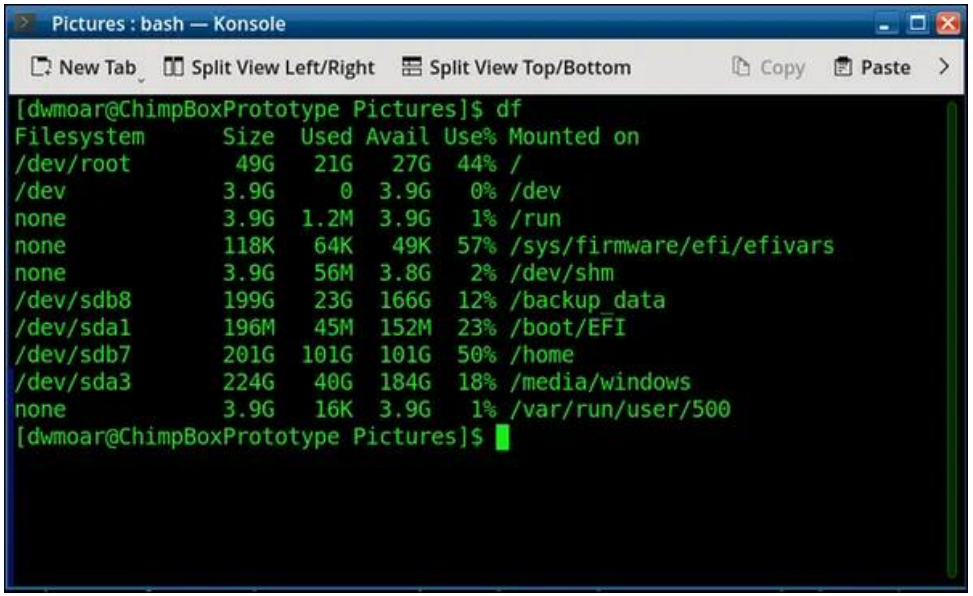


-T print filesystem typedf -l

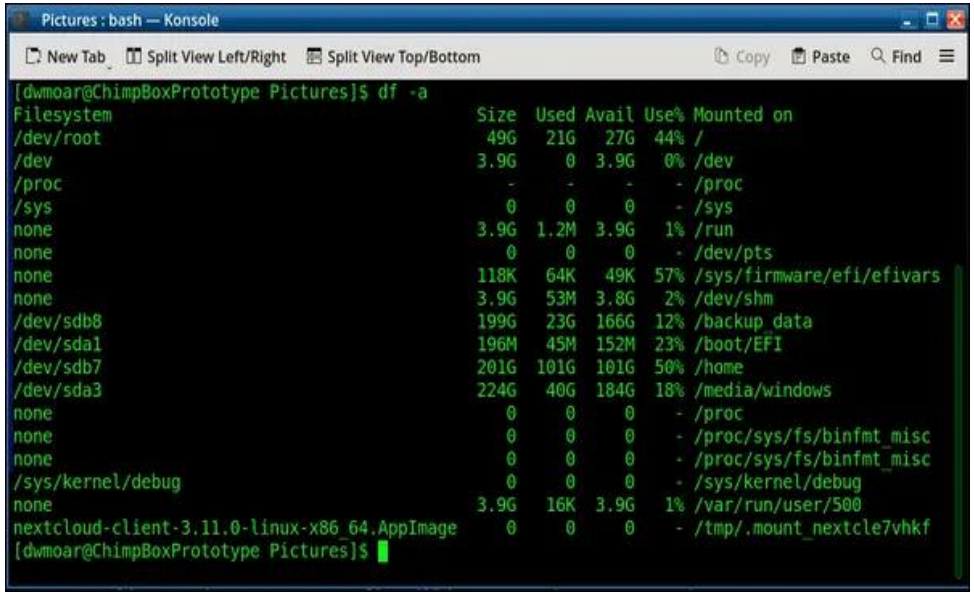
-i list inode information instead of block usage

Below are some examples of the df command with each argument used.

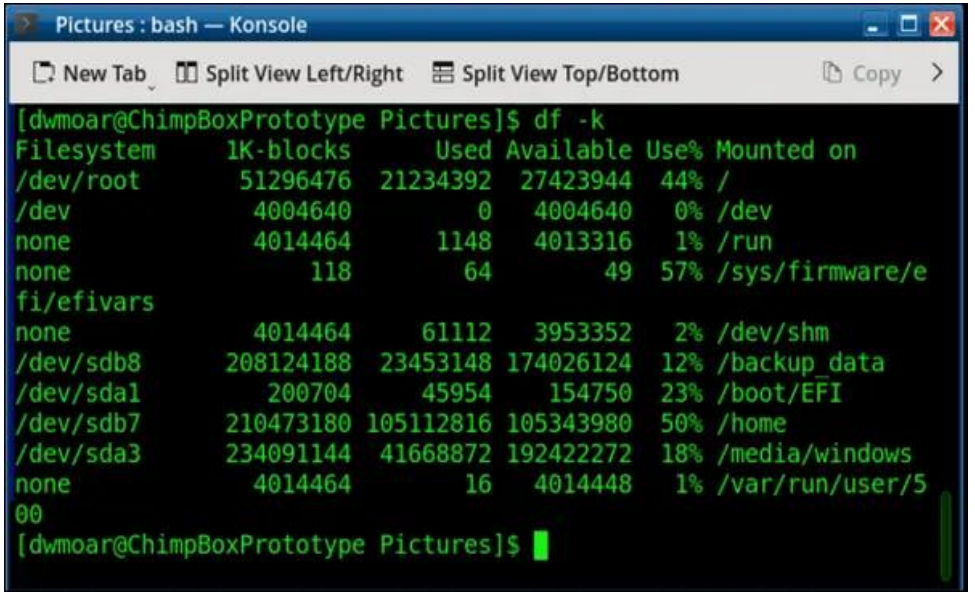
df



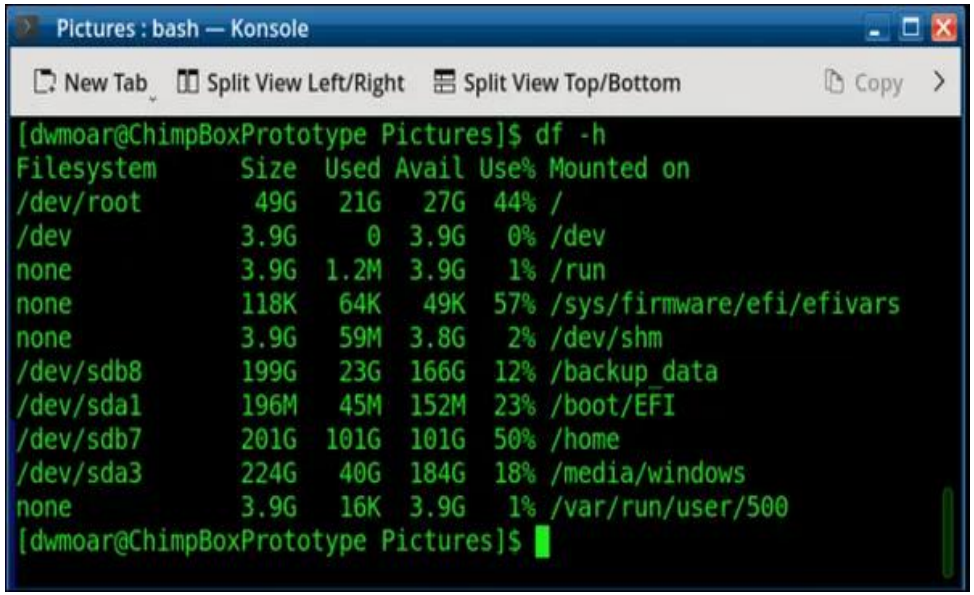
df -a



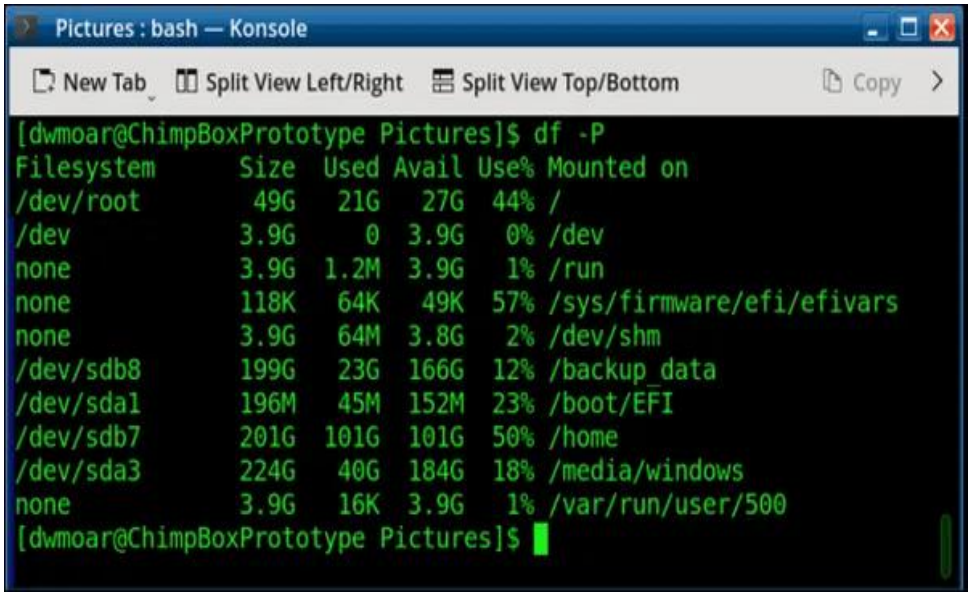
df -k



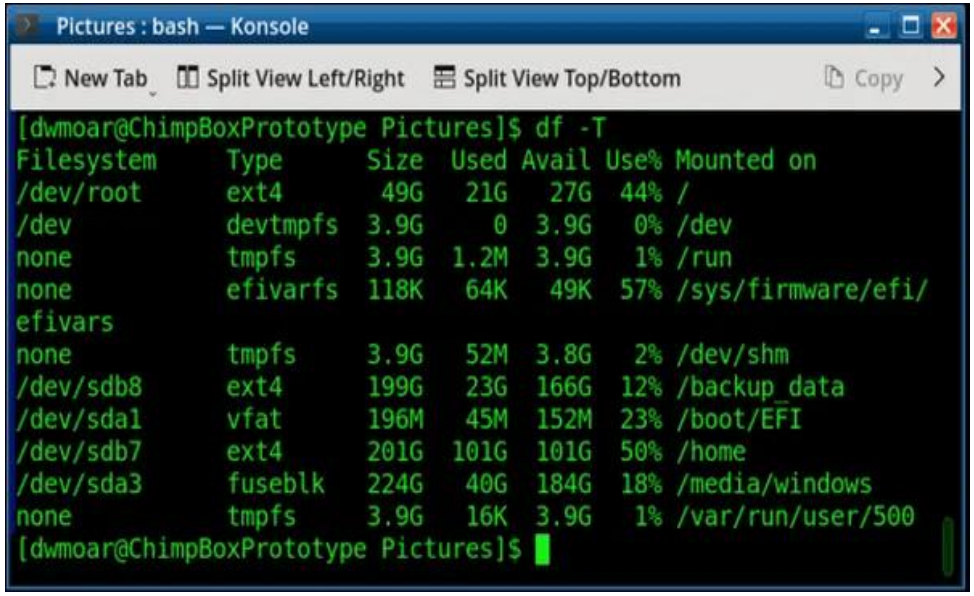
df -h



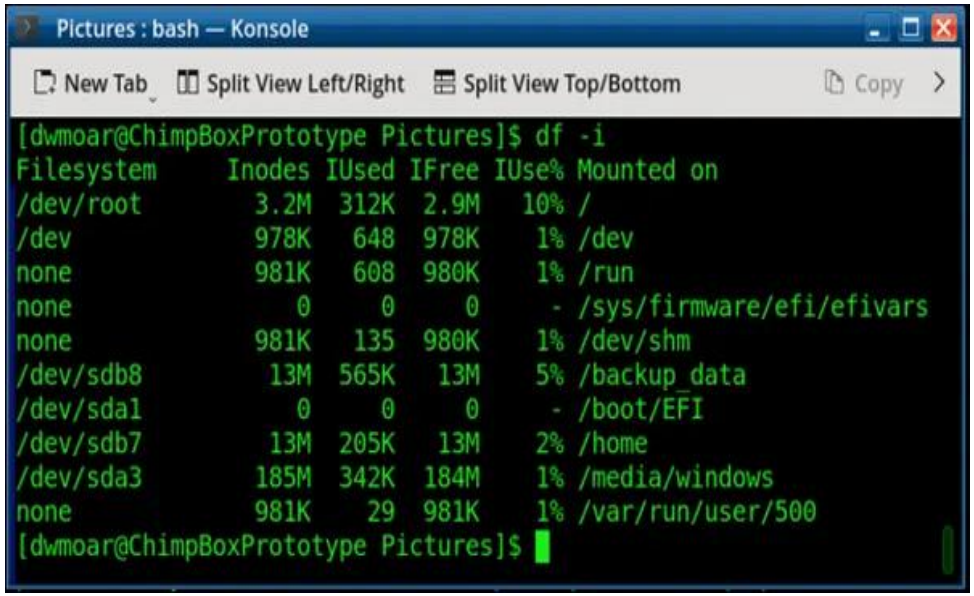
df -P



df -T



df -i



What is Filelight

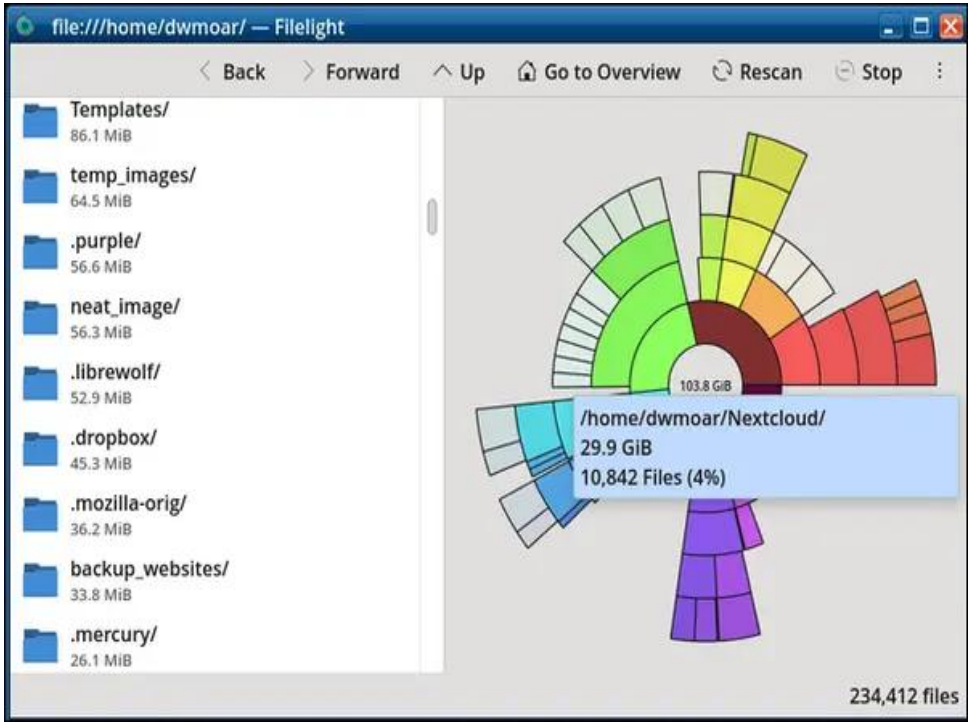
For those that prefer a graphical disk usage program.

Filelight allows you to quickly understand exactly where your disk space is being used by graphically representing your file system as a set of concentric segmented-rings. You can use it to locate hotspots of disk usage and then manipulate those areas using Dolphin or Konqueror.

You can find it in the menu -> File Tools -> Filelight.

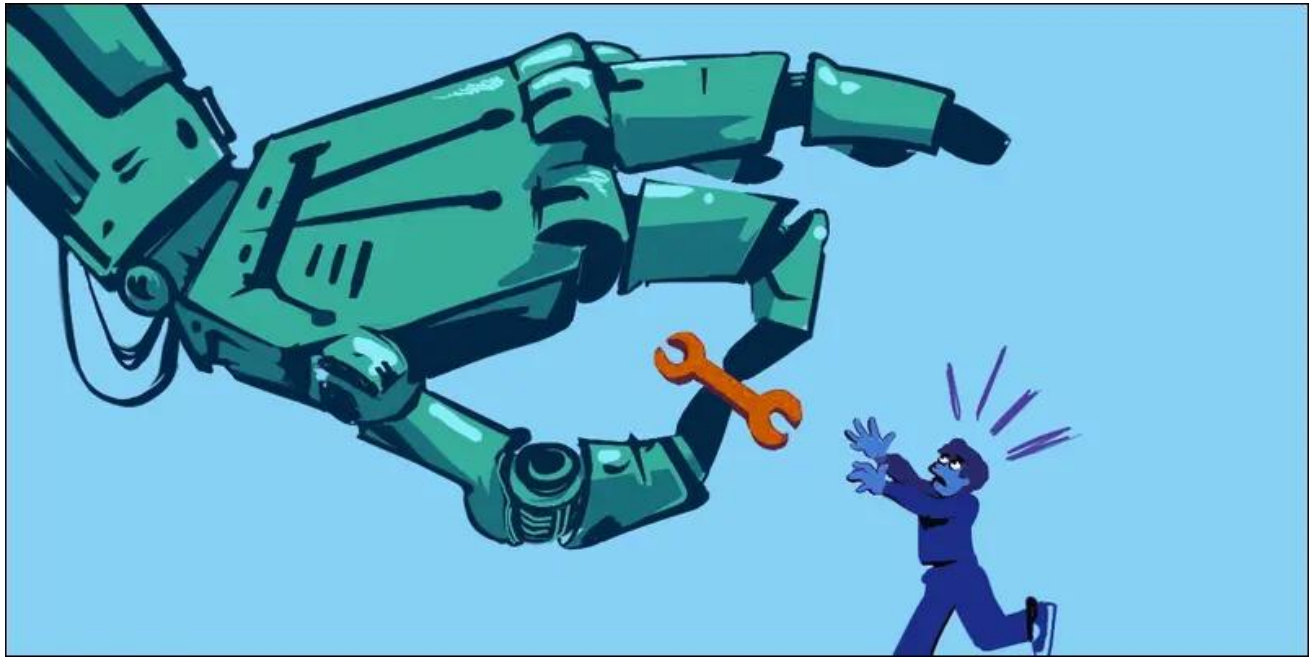
If you can not locate it there, you may need to be installed through synaptic package manager.

Here is an example of my hard drive looking at my Nextcloud folder:



You can view the original PCLinuxOS Wiki Knowledgebase article here.

Print Blocking Won't Work: Permission To Print, Part 2



by Cliff Braun
Electronic Frontier Foundation
Reprinted under Creative Commons license

Legislators across the U.S. are proposing laws to force "print blockers" on 3D printers sold in their states. This mandated censorware is doomed to fail for its intended purpose, but will still manage to hurt the professional and hobbyist communities relying on these tools.

3D printers are commonly used to repair belongings, decorate homes, print figurines, and so much more. It's not just hobbyists; 3D printers are also used professionally for parts prototyping and fixturing, small-batch manufacturing, and workspace organization. In rare cases, they've also been used to print parts needed for firearm assembly.

Many states have already banned manufacturing firearms using computer controlled machine tools, which are called "Computer Numerical Control or CNC machines," and 3D printers without a license. Recently proposed laws seek to impose technical limitations onto 3D printers (and in some cases, CNC machines) in the hope of enforcing this prohibition.

This is a terrible idea; these mandates will be onerous to implement and will lock printer users into vendor software, impose one-time and ongoing costs on both printer vendors and users, and lay the foundation for a 3D-print censorship platform to be used in other jurisdictions. We dive more into these issues in the first part of this series (reprinted elsewhere in this issue).

On a pragmatic level, however, these state mandates are just wishful thinking. Below, we dive into how 3D printing works, why these laws won't deter the printing of firearms, and how regular lawful use will be caught in the proposed dragnet.

How 3D Printers Work

To understand the impact of this proposed legislation, we need to know a bit about how 3D printers work. The most common printers work similarly to a computer-controlled hot glue gun on a motion platform; they follow basic commands to maintain temperature, extrude (push) plastic through a nozzle, and move a platform. These motions together build up layers to make a final "print." Modern 3D printers often offer more features like Wi-Fi connectivity or camera monitoring, but fundamentally they are very simple machines.

The basic instructions used by most 3D printers are called Geometric Code, or G-Code, which

specify very basic motions such as “move from position A to position B while extruding plastic.” The list of commands that will eventually print up a part are transferred to the printer in a text file thousands-to-millions of lines long. The printer dutifully follows these instructions with no overall idea of what it is printing.

While it is possible to write G-Code by hand for either a CNC machine or a 3D printer, the vast majority is generated by computer aided manufacturing (CAM) software, often called a “slicer” in 3D printing since it divides a 3D model into many 2D slices then generates motion instructions.

This same general process applies to CNC machines which use G-Code instructions to guide a metal removal tool. CNC machines have been included in previous prohibitions on firearm manufacturing and file distribution and are also targeted in some of these bills.

There are other types of 3D printers such as those that print concrete, resin, metal, chocolate and other materials using slightly different methods. All of these would be subject to the proposed requirements regardless of how unlikely doing harm with a gun made out of chocolate would be.

Part of a 173,490 line long G-Code file for a simple rectangular part.

How is Firearm Detection Supposed to Work?

Under these proposed laws, manufacturers of consumer 3D printers must ensure their printers only work with their software, and implement firearm detection algorithms on either the printer itself or in a slicer software. These algorithms must detect firearm files using a maintained database of existing models. Vendors of printers must then verify that printers are on the allow-list maintained by the state before they can offer them for sale.

Owners of printers will be guilty of a crime if they circumvent these intrusive scanning procedures or load alternative software, which they might do because their printer manufacturer ends support. Owners of existing

noncompliant 3D printers in regulated states will be unable to resell their printers on the secondary market legally.

What Will Actually Happen?

While the proposed laws allow for scanning to happen on either the printer itself or in the slicer software, the reality is more complicated.

The computers inside many 3D printers have very limited computational and storage ability; it will be impossible for the printer's computer to render the G-Code into a 3D model to compare with the database of prohibited files. Thus, the only way to achieve this through the machine would be to upload all printer files to a cloud comparison tool, creating new delays, errors, and unacceptable invasions of privacy.

Many vendors will instead choose to permanently link their printers to a specific slicer that implements firearm detection. This requires cryptographic signing of G-Code to ensure only authorized prints are completed, and will lock 3D printer owners into the slicer chosen by their printer vendor.

Regardless of the specifics of their implementation, these algorithms will interfere with 3D printers' ability to print other parts without actually stopping manufacture of guns. It takes very little skill for a user to make slight design tweaks to either a model or G-Code to evade detection. One can also design incomplete or heavily adorned models which can be made functional with some post-print alterations. While this would be pioneered by skilled users — like the ones who designed today's 3D printed guns — once the design and instructions are out there anyone able to print a gun today will be able to follow suit.

Firearm part identification features also impose costs onto 3D printer manufacturers, and hence their end consumers. 3D printer manufacturers must develop or license these costly algorithms and continuously maintain and update both the algorithm and the database of firearm models. Older printers that cannot comply will not be able to be resold in states where they are banned, creating additional E-waste.

While those wishing to create guns will still be able to do so, people printing other functional parts will likely be caught up in these algorithms, particularly for things like film props, kids' toys, or decorative models, which often closely resemble real firearms or firearm components.

What Are The Impacts of These Changes?

Technological restrictions on manufacturing tools' abilities are harmful for many reasons. EFF is particularly concerned with this regulation locking a 3D printer to proprietary vendor software. Vendors will be able to use this mandate to support only in-house materials, locking users into future purchases. Vendor slicer software is often based on out-of-date, open source software, and forcing users to use that software deprives them of new features or even use of their printer altogether if the vendor goes out of business. At worst, some of these bills will make it a misdemeanor to fix those problems and gain full control of your printer.

File-scanning frameworks required by this regulation will lay the foundation for future privacy and freedom intrusions. This requirement could be co-opted to scan prints for copyright violations and be abused similar to DMCA takedowns, or to suppress models considered obscene by a patchwork of definitions. What if you were unable to print a repair part because the vendor asserted the model was in violation of their trademark? What if your print was considered obscene?

Regardless of your position on current prohibitions on firearms, we should all fight back against this effort to force technological restrictions on 3D printers, and legislators must similarly abandon the idea. These laws impose real costs and potential harms among lawful users, lay the groundwork for future censorship, and simply won't deter firearm printing.

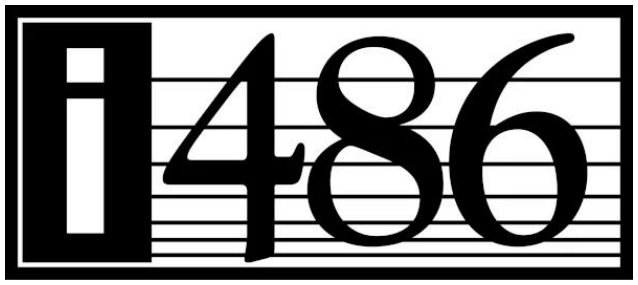
ICYMI: Linux 7.1 Kernel Phasing Out i486 CPU Support

by Paul Arnote (parnote)

Google is changing how Android users install apps from outside the Play Store, introducing a new process that aims to preserve the platform's sideloading feature, while making it more difficult for scammers to exploit it, according to an article from TechRepublic. The new installation process requires several steps that go far beyond a change in "settings". Users must first enable developer mode, a hidden feature typically used by programmers and advanced users that require navigating deep into system settings and performing specific actions that most everyday users would never encounter. From there, users must confirm they are not being pressured by someone else to disable security protections, then comes a required device restart and reauthentication step, which Google says is meant to cut off remote access tools or active calls that scammers might be using to monitor victims in real time. The next step is the mandatory 24-hour waiting period, which Google says is meant to protect users against one of scammers' most effective psychological tactics, the use of urgency. By forcing users to wait a full day before completing the process, the company hopes people will have time to reconsider decisions they might otherwise make under pressure. After the waiting period, users must reauthenticate, a measure intended to block remote access sessions or active scam calls that could influence their actions.

Want to use almost any computer without leaving a trace? According to an article from Lifehacker, you need to meet Tails (The Amnesic Incognito Live System), a Linux distro based on Debian that maximizes user security and privacy. It can run from a USB drive, which means you're able to plug it into any laptop or desktop with a free port (Mac or PC) and take it away again when you're done, with nothing left behind on the computer you borrowed. Tails also comes with anonymous internet browsing built in as standard, through the Tor browser and the Tor network that reroutes your online activity across several private web nodes. No one can tell who you are or where you are, and you can get online without all the tracking and monitoring that's usually associated with opening up a browser. It's perfect for hiding who you are online, and for dialing up your privacy and security protections to the absolute maximum—not to mention getting around state-level censorship, if that applies to you.

Do you still have that Gmail account from your younger years? You know, that one that reads JoeLovesSusan4Ever@gmail.com? Or maybe SchoolSUCKS@gmail.com? The ability to change that initial Gmail account name has started to roll out in the U.S., according to an article from ArsTechnica. You are limited to one email change every 12 months, and you can do so without creating a whole new account. The email and other data in your account remain untouched when you migrate to a new username, and you will continue to receive mail destined for your original address. You can also log in to your account using either username. Some Google products and third-party services that access Google data will continue to display your original email, but the new username becomes your primary Google account identifier. If you don't want to see that old handle anywhere, the only option is still to create a totally new account. There is a Google Support page for how to affect the change, here.



A patch queued into one of the development branches ahead of the upcoming Linux 7.1 merge window is set to finally begin the process of phasing out and ultimately removing Intel 486 CPU support from the Linux kernel, according to an article from Phoronix. Anyone still using an i486 CPU with an upstream Linux kernel would be incredibly rare and no known Linux distribution vendors are still shipping with i486 CPU support, but in case you are, you can continue to be running one of the existing Linux LTS kernel versions. Linus Torvalds recently commented that he's feeling like it's time for letting the Linux kernel go of its long-present i486 CPU support as there's "zero real reason" to keep it around and waste upstream Linux kernel development efforts.

A new scientific study from the University of California, San Francisco (UCSF) has come to the startling conclusion that a single protein is the catalyst for cognitive dysfunction — and the damage it causes can be reversed, according to an article from Inc. Scientists at UCSF's Bakar Aging Research Institute examined activity in the hippocampus, the brain's command center for learning and memory. Comparing young and old mice, the researchers discovered that older brains, unlike younger ones, were flooded with the FTL1 protein. To figure out whether the protein was actually the culprit or just another byproduct of the aging process, they elevated FTL1 levels in young mice, whose brains soon began to look and act old. Their neurons quit branching out into complex networks, shrinking instead into stubby extensions that no longer communicated with the efficiency of youth. But it was the next step in their experiment that ended in a revelation. Lowering the protein levels in older mice didn't just slow or stop the fraying of the older brain. Clearing out the FTL1 clutter helped rebuild lost connections in the hippocampus and literally healed existing damage. The proof: the treated mice soon scored "significantly better" on memory tests.

From an article on Tom's Hardware, **Via Licensing Alliance (Via LA), the patent pool administrator for H.264/AVC, quietly restructured its streaming license fees recently, replacing a flat $100,000 annual cap with a tiered system that tops out at $4,500,000 per year for the largest platforms**, according to a *Streaming Media* report published on March 17. The change applies only to previously unlicensed implementers seeking a new license in 2026 or later, with all companies that held an active AVC license as of the end of 2025 retaining their original terms. The new hike for H.264 comes in the wake of disastrous increases in HEVC/H.265 fees that led to widespread issues spanning the globe, including Asus and MSI laptops being banned in Germany.

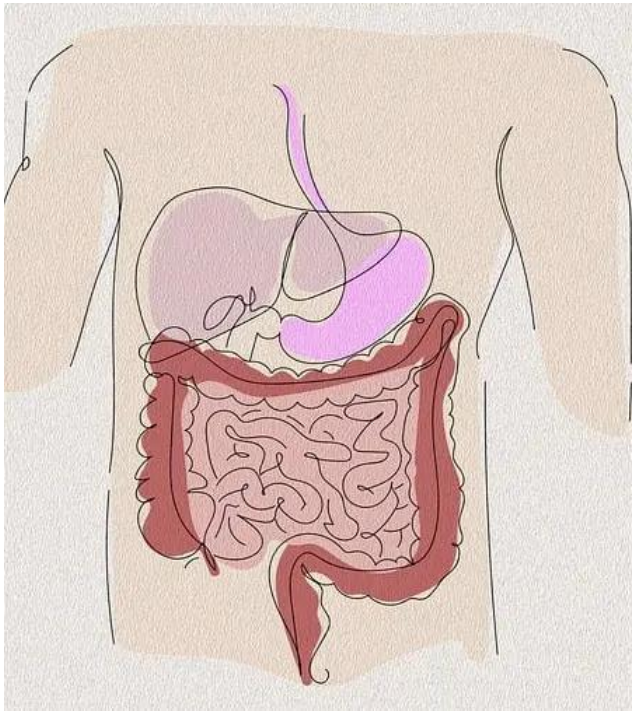


Image by Bianca Van Dijk from Pixabay

Researchers have discovered that a particular protein in the gut is fighting an important battle against the spread of bacteria, according to an article from ScienceAlert. Through a combination of protective effects, the protein could in the future be used in treatments for conditions such as inflammatory bowel disease. The protein, intelectin-2, isn't new to scientists, but its role in the gastrointestinal (GI) tract hasn't been clear. Intelectin-2 is part of a group of proteins called lectins, which work by binding to specific sugar molecules. A team led by researchers at MIT has now found that intelectin-2 works in two ways: First, it links mucus molecules lining the GI tract, strengthening the mucus barrier that

protects intestinal tissues. But if there's a breach, intelectin-2 also recognizes and traps a variety of bacterial cells, either inhibiting their growth or killing them off completely. So intelectin-2's role is a combination of defense and offense activities that protect bodily health.

Every month I receive validation that I made the right decision to stop using Google Chrome due to the (literally) monthly security updates to plug critical vulnerabilities (among other reasons). It looks like the trend continues ad nauseam, **since just days after Google started rolling out a high-risk security update for Chrome users, the technology giant has now issued a new alert following the confirmation of a new zero-day exploit observed in the wild**, according to an article from Forbes. This means that hackers already have a head start when it comes to using CVE-2026-5281 in attacks against the 3.5 billion users of the world's most popular web browser. The good news is that Google has commenced another security update distribution to address this high-severity vulnerability, along with a staggering 20 others. The bad news is that it could take days, or even weeks, to reach you, according to the Google announcement. Thankfully, however, there is a way to ensure that your Chrome browser gets that Google security update right now, and you can scroll to the end of the linked article for step-by-step instructions.

A new report dubbed "BrowserGate" warns that Microsoft's LinkedIn is using hidden JavaScript scripts on its website to scan visitors' browsers for installed extensions and collect device data, according to an article from BleepingComputer. According to a report by Fairlinked e.V., which claims to be an association of commercial LinkedIn users, Microsoft's platform injects JavaScript into user sessions that checks for thousands of browser extensions and links the results to identifiable user profiles. The author claims that this behavior is used to collect sensitive personal and corporate information, as LinkedIn accounts are tied to real identities, employers, and job roles. "LinkedIn scans for over 200 products that directly compete with its own sales tools, including Apollo, Lusha, and ZoomInfo. Because LinkedIn knows each user's employer, it can map which companies use which competitor products. It is extracting the customer lists of thousands of software companies from their users' browsers without anyone's knowledge,' the report says.

Image by Daniele Liberatori from Pixabay

Rates of diabetes are lower in high-altitude locations, but researchers have been unsure why. **Now, a new study in mice reveals a possible explanation: Red blood cells, which play a pivotal role in transporting oxygen throughout the body, may lower blood sugar by converting glucose into a compound that helps release oxygen into tissues**, according to an article from LiveScience. If the results can be replicated in people, they also hint that drugs in early-stage development could potentially mimic this pathway. "The work highlights the important role that red blood cells can play in diabetes regulation," study lead author Isha Jain, a biochemist at the Gladstone Institutes and the University of California, San Francisco, told Live Science. "That's the concept to be targeted in the future."

Scientists have cured type 1 diabetes in mice, without long-term immune suppression, according to another article from LiveScience. In type 1 diabetes, the immune system attacks insulin-producing cells, and replacing them with transplanted cells from donors has historically required people to take strong immunosuppressants for life, which severely limited the reach of such transplants. But in a new study, researchers created a "chimeric," or blended immune system that contains elements of both the recipient's and the donor's immune systems. This enabled mice to tolerate a transplant of insulin-producing cells without long-term immune suppression. Much more research is needed before this kind of treatment could be available to patients in a clinic, and keeping the blended immune system balanced is tricky. But if extensive follow-up testing in humans shows the transplantation process is safe and durable, it could offer an avenue for reversing the potentially deadly disease. "This is

potentially a way to cure diabetes," Dr. John DiPersio, an oncologist at Washington University in St. Louis who researches cellular therapy but was not involved in the study, told Live Science. "It does represent, in theory, a big step forward."

Early observations from the Vera C. Rubin Observatory have already revealed more than 11,000 previously unknown asteroids, reshaping our view of the solar system and offering a striking preview of what's to come once full science operations begin, according to an article from Space.com. The discovery, made using preliminary data, demonstrates Rubin's ability to scan the sky quickly and deeply. Even during limited early observations, the telescope has detected thousands of moving objects in just days, far outpacing traditional asteroid surveys, according to a statement from the NSF NOIRLab.

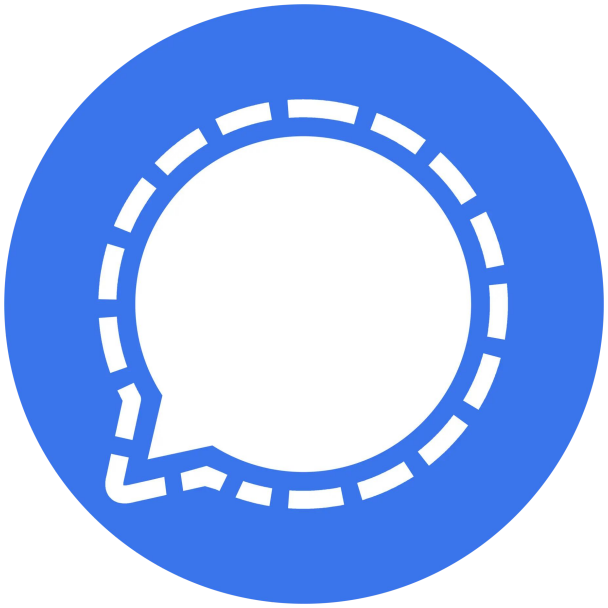
Image by Paweł from Pixabay

The top official at NASA says that the chance of alien existence is a factor in how the US space agency plans its missions, according to an article from The Guardian. Speaking on April 5, 2026, NASA administrator Jared Isaacman told CNN's Meet the Press that investigating the existence of alien life "goes to the heart of many things that we do at NASA", adding: "Our job here is to go out and try and unlock the secrets of the universe." One of the questions, he said, is "are we alone? The question would say that is inherent in every one of our scientific endeavors, our exploration endeavors." Isaacman pointed to a potential moon base on the South Pole of the moon that would incorporate telescopes "that will help us continue this great search". But the official qualified his comments, offering that he had been to space twice and "didn't encounter any aliens up there. I have not seen anything to suggest that we have been visited by any intelligent life forms out there." But, he added, "when you think about it, we got 2tn galaxies out there. Who knows how many star systems within each of it? I would say the odds that we will find something at some point to suggest that we are not alone are pretty high."

Millions of Android users are now eligible to claim some cash from Google as part of a $135 million settlement, according to an article from Lifehacker. This settlement is part of a class-action lawsuit filed earlier this year alleging that Google collected unnecessary data from Android users over cellular networks. Payouts are capped at $100, though the total could be significantly less if the estimated 100 million class members receive equal amounts. Just in case you're wondering, I wouldn't expect this to change any egregious behavior by Google. Since they are a multi-billion dollar corporation, this paltry amount isn't a punishment. Instead, it's "just the cost of doing business."

Chrome is rethinking how you browse with new vertical tabs and a cleaner reading mode to cut out clutter and boost focus, according to an article from TechRepublic. Google announced a new Chrome update on April 7. But instead of its routine fixes, the tech giant introduces two notable shifts that change how users navigate the browser. For the first time since its launch, Chrome is changing how tabs are positioned, giving users the option to switch to a vertical layout or stick with the traditional horizontal view. In addition to tab repositioning, Chrome introduced a better way to focus while reading: a reading mode that instantly cuts off distracting media, improving overall productivity.

A magazine just isn't a magazine without articles to fill the pages. If you have article ideas, or if you would like to contribute articles to the PCLinuxOS Magazine, send an email to: pclinuxos.mag@gmail.com We are interested in general articles about Linux, and (of course), articles specific to PCLinuxOS.



The FBI was able to pull Signal messages from a defendant's iPhone — even though that user had deleted the app, according to an article from Lifehacker. The FBI tapped into the iPhone's notification database, where they found alerts containing incoming messages the user received. Signal has a setting that can block this vulnerability, but it has to be enabled manually. You might have heard about Signal, the encrypted chat app. But while the app is no alternative to a dedicated SCIF, it is a good option for the rest of us to communicate more securely. Signal uses end-to-end encryption (E2EE), which, very simply, means that messages are "scrambled" in transit, and can only be "unscrambled" by the sender and the recipient or recipients. If you're in a Signal chat, you'll be able to read incoming messages just like you would any other chat app — if you're an attacker, and intercept that message, all you'll find is a jumble of code. E2EE makes it difficult for anyone without your unlocked device (or your unlocked Signal app) to read your Signal message difficult, but not impossible. That's part of the reason the chat app is no option for government officials (though no third-party chat app could be). But it's also a good reminder that no matter who you are, your secure chats are not impervious to outside forces. If someone wants to break into your chats, they might find a way to do so.

Cybersecurity researchers have discovered a new campaign in which a cluster of 108 Google Chrome extensions has been found to communicate with the same command-and-control (C2) infrastructure with the goal of collecting user data and enabling browser-level abuse by injecting ads and arbitrary JavaScript code into every web page visited, according to an article from Hacker News. According to Socket, the extensions (complete list here) are published under five distinct publisher identities – Yana Project, GameGen, SideGames, Rodeo Games, and InterAlt – and have collectively amassed about 20,000 installs in the Chrome Web Store. Of these, 54 add-ons steal Google account identity via OAuth2, 45 extensions contain a universal backdoor that opens arbitrary URLs as soon as the browser is started, and the remaining ones engage in a variety of malicious behaviors.

The last week of the release continued the same "lots of small fixes" trend, but it all really does seem pretty benign, so I've tagged the final 7.0 and pushed it out, reported Linus Torvalds on the Linux Kernel Mailing List (LKML.org) on April 12, 2026. I suspect it's a lot of AI tool use that will keep finding corner cases for us for a while, so this may be the "new normal" at least for a while. Only time will tell. Anyway, this last week was a little bit of everything: networking (core and drivers), arch fixes, tooling and self tests, and various random fixes all over the place. Let's keep testing, and obviously tomorrow the merge window for 7.1 opens. I already have four dozen pull requests pending - thank you to all the early people.

Image by Shelley Evans from Pixabay

New research has linked levels of vitamin D in midlife with toxic tangles of tau protein that accumulate in the brains of those with Alzheimer's disease, according to an article from ScienceAlert. A statistical analysis of blood samples and brain scans from 793 adults

showed that the more vitamin D in someone's system in middle age, the lower the amount of tau protein tangles they tended to have years later. The finding comes from an international team of researchers, and while it doesn't prove direct cause and effect, it suggests an association that's worth looking at.

The x86/asm changes merged yesterday (April 14) for the Linux 7.1 kernel with a few low-level improvements, according to an article from Phoronix. Uros Bizjak worked out a few of the x86/x86_64 Assembly improvements for the Linux 7.1 kernel. For the most part it's uneventful work this cycle, but there are two patches for removing some unnecessary memory clobbers. Avoiding the memory clobbers from the inline Assembly code can be useful for minor impact to better instruction scheduling and register allocation. The memory clobbers act as a read/write memory barrier to prevent the compiler from reordering memory loads/stores from the inline Assembly statement and to flush any values cached in registers back to memory as well as reloading values cached in registers that may have been changed in the Assembly code.

Fred Hutch Cancer Center researchers have made a significant advance in the effort to block Epstein Barr virus (EBV), a widespread infection that affects about 95% of people worldwide and is linked to several cancers, neurodegenerative conditions, and other long-term illnesses, according to an article from ScienceAlert. By working with mice engineered to produce human antibodies, the team created new monoclonal antibodies designed to stop the virus from attaching to and entering human immune cells. The findings, published in Cell Reports Medicine, show that one of these antibodies was able to completely prevent infection in mice with human-like immune systems when exposed to EBV.

Image by CSSAuthor

Picture this: your brain is a high-performance engine. Over decades, it doesn't just wear down, it also starts to run hot, according to an article from Texas A&M. Tiny "fires" of inflammation smolder deep within the brain's memory center, creating a persistent brain fog that makes it harder to think, form new memories or even adapt to new environments, all the while increasing the risk to disorders like Alzheimer's disease. Scientists call this slow burn "neuroinflammaging," and for decades it was thought to be the inevitable price of growing older. Until now. A landmark study from researchers at the Texas A&M University Naresh K. Vashisht College of Medicine suggests the inflammatory tide responsible for brain aging and brain fog might actually be reversible. And the solution doesn't involve brain surgery, but a simple nasal spray. Led by Dr. Ashok Shetty, university distinguished professor and associate director of the Institute for Regenerative Medicine, along with senior research scientists Dr. Madhu Leelavathi Narayana and Dr. Maheedhar Kodali, the team developed a nasal spray that, with just two doses, dramatically reduced brain inflammation, restored the brain's cellular power plants and significantly improved memory. The most surprising part? It all happened within weeks and lasted for months.

Researchers have found a way to make the immune system's T cells far more effective at attacking cancer. By blocking a protein known as Ant2, they were able to change how these cells produce and use energy, essentially rewiring their internal power supply, according to an article from ScienceDaily. This transformation makes T cells more active, more durable, and better equipped to destroy tumors. The discovery points to new treatment strategies that strengthen the body's natural defenses, offering a more precise approach to cancer therapy. A new study suggests a path toward next-generation cancer treatments by training the immune system to respond more efficiently and aggressively. The research was led by PhD student Omri Yosef and Prof. Michael Berger from the Faculty of Medicine at Hebrew University, working with Prof. Magdalena

Huber of Philipps University of Marburg and Prof. Eyal Gottlieb of the University of Texas MD Anderson Cancer Center. Together, the international team found that adjusting how immune cells handle energy can greatly improve their ability to eliminate cancer. At the center of this work is a key idea: when T cells, which play a central role in immune defense, are forced to alter how they convert energy, they become much better at detecting and attacking cancer cells.

A largely overlooked plant compound found in common fruits and vegetables is drawing new scientific attention for its potential effects on aging and brain health, according to an article from SciTechDaily. A little-known nutrient found in everyday fruits and vegetables may be doing far more in the body than scientists once believed. Researchers from the University of Seville and the University of Kent report that phytoene, a colorless carotenoid present in foods like tomatoes, carrots, oranges, and peppers, can extend lifespan and protect against key processes linked to Alzheimer's disease, at least in a widely used laboratory model. See? Your momma was right … always eat your fruits and vegetables.

Image from Pixabay

The International Diabetes Federation (IDF) officially recognized a fifth form of diabetes in 2025, after decades of controversy, according to an article from Science Alert. It's now urging other health authorities, like the World Health Organization (WHO), to follow suit. Type 5 diabetes is rarely discussed or researched, and yet it is thought to impact up to 25 million people worldwide, especially those in low- and middle-income nations where access to medical care is limited. It was first described in 1955 in Jamaica, then forgotten about for many years. Even once it was acknowledged by the WHO in the 1980s, the diagnosis created controversy. For going on seven decades, scientists have debated whether type 5 diabetes exists at all, and in 1999, WHO withdrew the classification due to a lack of evidence. There has been little agreement on how to diagnose type 5 diabetes or how to treat it. Type 1 diabetes is an autoimmune condition that destroys the pancreas's ability to produce insulin. Type 2 diabetes is an insensitivity to insulin due to diet and lifestyle. Type 3c diabetes is commonly caused by damage to the pancreas. Gestational diabetes is triggered by hormonal changes during pregnancy. Type 5 diabetes seems to stem from nutrient deficiency.

Scientists at Stanford Medicine have identified a naturally occurring molecule that appears to mimic some of the weight loss effects of semaglutide, the drug widely known as Ozempic, according to an article from Science Daily. In animal studies, the molecule reduced appetite and body weight while avoiding several common side effects such as nausea, constipation, and muscle loss. The molecule, called BRP, works through a different but related biological pathway and activates distinct groups of neurons in the brain. This suggests it may offer a more precise way to control appetite and metabolism.

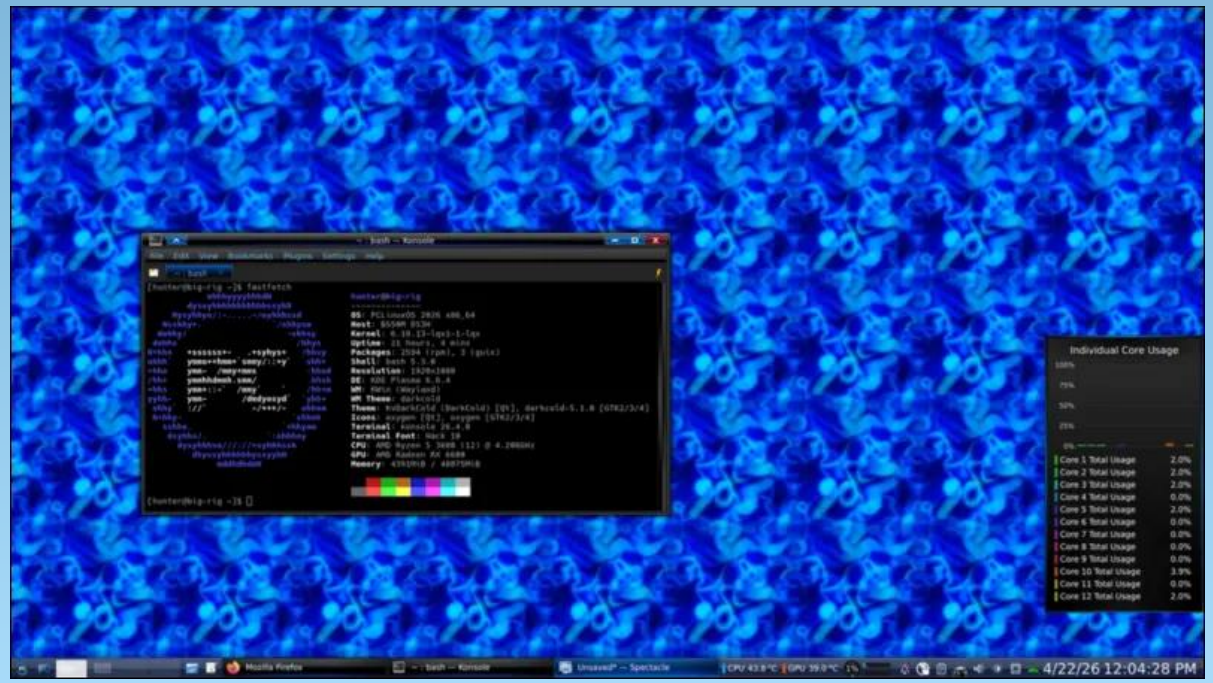
Carl Richell, the CEO and founder of Linux hardware vendor System76, shared today on fediverse that the upcoming Colorado Age Attestation bill has been amended to exclude Linux distros and Open Source apps, according to an article from 9To5Linux. As you may know, several US states are discussing a **Digital Age Assurance Act** law that mandates that operating system providers and application developers implement age verification measures to protect minors online, which may require users to input their birthdate during the initial setup. System76 CEO Carl Richell has been in

discussions with Colorado Senator Matt Ball, the co-author of the Colorado OS age attestation bill, to exclude open source software from the upcoming age attestation bill, and today he received an updated version of the bill that does exactly that. The amended age attestation bill apparently excludes all open source operating systems and applications, as well as code repositories like GitHub and GitLab, and containers like Docker or Podman.

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 hunter0one, on April 22, 2026, running KDE.

The Government Uses Targeted Advertising To Track Your Location

by Lena Cohen and Hudson Hongo
Electronic Frontier Foundation

We've all had the unsettling experience of seeing an ad online that reveals just how much advertisers know about our lives. You're right to be disturbed. Those very same online ad systems have been used by the government to warrantlessly track peoples' locations, new reporting has confirmed.

For years, the internet advertising industry has been sucking up our data, including our location data, to serve us "more relevant ads." At the same time, we know that federal law enforcement agencies have been buying up our location data from shady data brokers that most people have never heard of.

Now, a new report gives us direct evidence that Customs and Border Protection (CBP) has used location data taken from the internet advertising ecosystem to track phones. In a document uncovered by 404 Media, CBP admits what we've been saying for years: The technical systems powering creepy targeted ads also allow federal agencies to track your location.

The document acknowledges that a program by the agency to use "commercially available marketing location data" for surveillance drew from the process used to select the targeted ads shown to you on nearly every website and app you visit. In this blog post, we'll tell you what this process is, how it can and is being used for state surveillance, and what can be done about it —by individuals, by lawmakers, and by the tech companies that enable these abuses.

Advertising Surveillance Enables Government Surveillance

The online advertising industry has built a massive surveillance machine, and the government can co-opt it to spy on us.

In the absence of strong privacy laws, surveillance-based advertising has become the norm online. Companies track our online and offline activity, then share it with ad tech companies and data brokers to help target ads. Law enforcement agencies take advantage of this advertising system to buy information about us that they would normally need a warrant for, like location data. They rely on the multi-billion-dollar data broker industry to buy location data harvested from people's smartphones.

We've known for years that location data brokers are one part of federal law

enforcement's massive surveillance arsenal, including immigration enforcement agencies like CBP and Immigration and Customs Enforcement (ICE). ICE, CBP and the FBI have purchased location data from the data broker Venntell and used it to identify immigrants who were later arrested. Last year, ICE purchased a spy tool called Webloc that gathers the locations of millions of phones and makes it easy to search for phones within specific geographic areas over a period of time. Webloc also allows them to filter location data by the unique advertising IDs that Apple and Google assign to our phones.

But a document recently obtained by 404 Media is the first time CBP has acknowledged the location data it buys is partially sourced from the system powering nearly every ad you see online: real-time bidding (RTB). As CBP puts it, "RTB-sourced location data is recorded when an advertisement is served."

Even though this document is about a 2019-2021 pilot use of this data, CBP and other federal agencies have continued to purchase and use commercially obtained location data. ICE has purchased location tracking tools since then and recently requested information on "Ad Tech" tools it could use for investigations.

The CBP document acknowledges two sources of location data that it relies on: software development kits (SDKs) and RTB, both methods of location-tracking that EFF has written about before. Apps for weather, navigation, dating, fitness, and "family safety" often request location permissions to enable key features. But once an app has access to your location, it could share it with data brokers directly through SDKs or indirectly (and often without the app developers' knowledge) through RTB. Data brokers can collect location data from SDKs that they pay developers to put in their apps. When relying on RTB, data brokers don't need any direct relationship with the apps and websites they're collecting location data from. RTB is facilitated by ad companies that are already plugged into most websites and apps.

How Real-Time Bidding Works

RTB is the process by which most websites and apps auction off their ad space. Unfortunately, the milliseconds-long auctions that determine which ads you see also expose your information, including location data, to thousands of companies a day. At a high-level, here's how RTB works:

1. The moment you visit a website or app with ad space, it asks an ad tech company to determine which ads to display for you.

2. This ad tech company packages all the information they can gather about you into a "bid request" and broadcasts it to thousands of potential advertisers.

3. The bid request may contain information like your unique advertising ID, your GPS coordinates, IP address, device details, inferred interests, demographic information, and the app or website you're visiting. The information in bid requests is called "bidstream data" and typically includes identifiers that can be linked to real people.

4. Advertisers use the personal information in each bid request, along with data profiles they've built about you over time, to decide whether to bid on the ad space.

5. The highest bidder gets to display an ad for you, but advertisers (or the adtech companies that represent them) can collect your bidstream data regardless of whether or not they bid on the ad space.

A key vulnerability of real-time bidding is that while only one advertiser wins the auction, all participants receive data about the person who would see their ad. As a result, anyone posing as an ad buyer can access a stream of sensitive data about billions of individuals a day. Data brokers have taken advantage of this vulnerability to harvest data at a staggering scale. For example, the FTC found that location data broker Mobilewalla collected data on over a billion people, with an estimated 60% sourced from RTB auctions. Leaked data from another location data broker, Gravy Analytics, referenced thousands of apps, including Microsoft apps, Candy Crush, Tinder, Grindr, MyFitnessPal, pregnancy trackers and religious-focused apps. When confronted, several of these apps' developers said they had never heard of Gravy Analytics.

As Venntel, one of the location data brokers that has sold to ICE, puts it, "Commercially available bidstream data from the advertising

ecosystem has long been one of the most comprehensive sources of real-time location and device data available." But the privacy harms of RTB are not just a matter of misuse by individual data brokers. RTB auctions broadcast the average person's data to thousands of companies, hundreds of times per day, with no oversight of how this information is ultimately exploited. Once your information is broadcast through RTB, it's almost impossible to know who receives it or control how it's used.

What You Can Do To Protect Yourself

Revelations about the government's exploitation of this location data shows how dangerous online tracking has become, but we're not powerless. Here are two basic steps you can take to better protect your location data:

1. **Disable your mobile advertising ID** (see instructions for iPhone/Android). Apple and Google assign unique advertising IDs to each of their phones. Location data brokers use these advertising IDs to stitch together the information they collect about you from different apps.

2. **Review apps you've granted location permissions to.** Apps that have access to your location could share it with other companies, so make sure you're only granting location permission to apps that really need it in order to function. If you can't disable location access completely for an app, limit it to only when you have the app open or only approximate location instead of precise location.

For more tips, check out EFF's guide to protecting yourself from mobile-device based location tracking. Keep in mind that the security plan that's best for you will vary in different situations. For example, you may want to take stronger steps to protect your location data when traveling to a sensitive location, like a protest.

What Tech Companies and Lawmakers Must Do

Legislators and tech companies must act so that individuals don't bear the burden of defending their data every time they use the internet.

Ad tech companies must reckon with their role in warrantless government surveillance, among other privacy harms. The systems they built for targeted advertising are actively used to track people's location. The best way to prevent online ads from fueling surveillance is to stop targeting ads based on detailed behavioral profiles. Ads can still be targeted contextually—based on the content people are viewing—without collecting or exposing their sensitive personal information. Short of moving to contextual advertising, tech companies can limit the use of their systems for government location tracking by:

- **Stopping the use of precise location data for targeted advertising.** Ad tech companies facilitating ad auctions can and should remove precise location data from bid requests. Ads can be targeted based on people's coarse location, like the city they're in, without giving data brokers people's exact GPS coordinates. Precise location data can reveal where we work, where we live, who we meet, where we protest, where we worship, and more. Broadcasting it to thousands of companies a day through RTB is dangerous.

- **Removing advertising IDs from devices, or at minimum, disabling them by default.** Advertising IDs have become a linchpin of the data broker economy and are actively used by law enforcement to track people's location. Advertising IDs were added to phones in 2012 to let companies track you, and removing them is not a far-fetched idea. When Apple forced apps to request access to people's advertising IDs starting in 2021 (if you have an iPhone you've probably seen the "Ask App Not to Track" pop-ups), 96% of U.S. users opted out, essentially disabling advertising IDs on most iOS devices. One study found that iPhone users were less likely to be victims of financial fraud after Apple implemented this change. **Google should follow Apple's lead and disable advertising IDs by default.**

Lawmakers also need to step up to protect their constituents' privacy. We need strong, federal privacy laws to stop companies from spying on us and selling our personal information. EFF advocates for data privacy legislation with teeth and a ban on ad targeting based on online behavioral profiles, as it creates a financial incentive for companies to track our every move.

Legislators can and must also close the "data broker loophole" on the Fourth Amendment. Instead of obtaining a warrant signed by a judge, law enforcement agencies can just buy location data from private brokers to find out where you've been. Last year, Montana became the first state in the U.S. to pass a law blocking the government from buying sensitive data it would otherwise need a warrant to obtain. And in 2024, Senator Ron Wyden's EFF-endorsed Fourth Amendment is Not for Sale Act passed the House before dying in the Senate. Others should follow suit to stop this end-run around constitutional protections.

Online behavioral advertising isn't just creepy–it's dangerous. It's wrong that our personal information is being silently harvested, bought by shadow-y data brokers, and sold to anyone who wants to invade our privacy. This latest revelation of warrantless government surveillance should serve as a frightening wakeup call of how dangerous online behavioral advertising has become.

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.

PCLinuxOS

Tip Top Tips: Using DNF Package Manager With Openbox

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 is from **kalwisti**.

I've been able to get the new DNF Package Manager working on Openbox. There were only a few (simple) adjustments needed.

Brief instructions are below for Openbox users who might be interested in trying this.

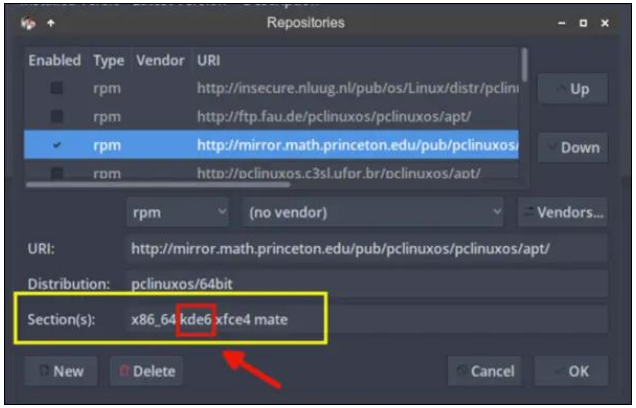
I had installed DNF Package Manager ["**DNF PM**"] but when I ran it for the first time, there were approx. 35 packages listed as upgradable (most of them were KDE-related). However, DNF PM could not successfully upgrade those packages because there were conflicts between the old/obsolete kde5 packages and the new kde6 packages.

The root of the problem was that my "**Section(s):**" listing in Synaptic included the now missing kde5 section. (KDE 5 is no longer supported, so that section of the repository is gone.)

In my case, the main offending programs were: Flameshot, Ksnip and KolourPaint (and their dependencies). To fix this, I did the following:

First, I removed (via Synaptic): Flameshot, Ksnip, KolourPaint.

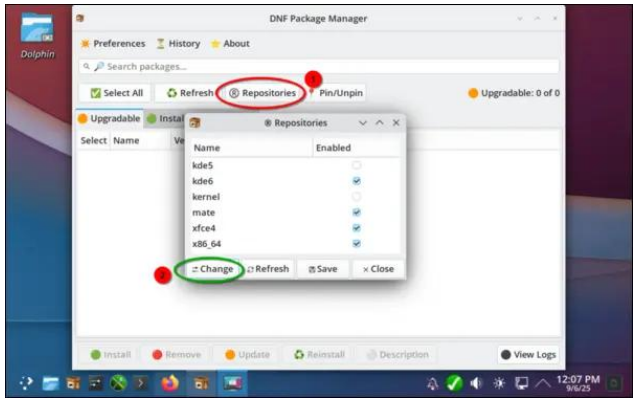
In Synaptic, I changed the kde5 section to kde6:

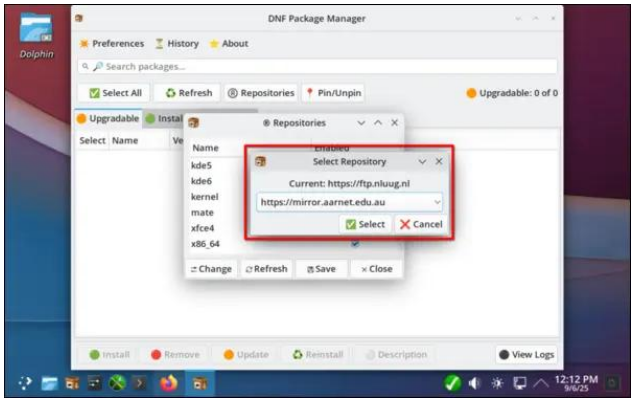


I updated the system via Synaptic.

Via Synaptic, install the **dnf-package-manager** package. It will install approx. 20 additional packages as dependencies.

I opened DNF PM and changed its default repo mirror (NLUUG) to my preferred mirror in the US (Princeton University).





Important: Do not disable any of the default **Section** settings in the repository list. In other words, do **not** disable the kde6 Section and do **not** select/tick kde5 to activate it.

I opened the DNF Package Manager and refreshed the package list. No more pending updates were shown. I installed Flameshot via DNF PM. The transaction succeeded with no problems. Both package managers are working smoothly now, but I am primarily using DNF PM.

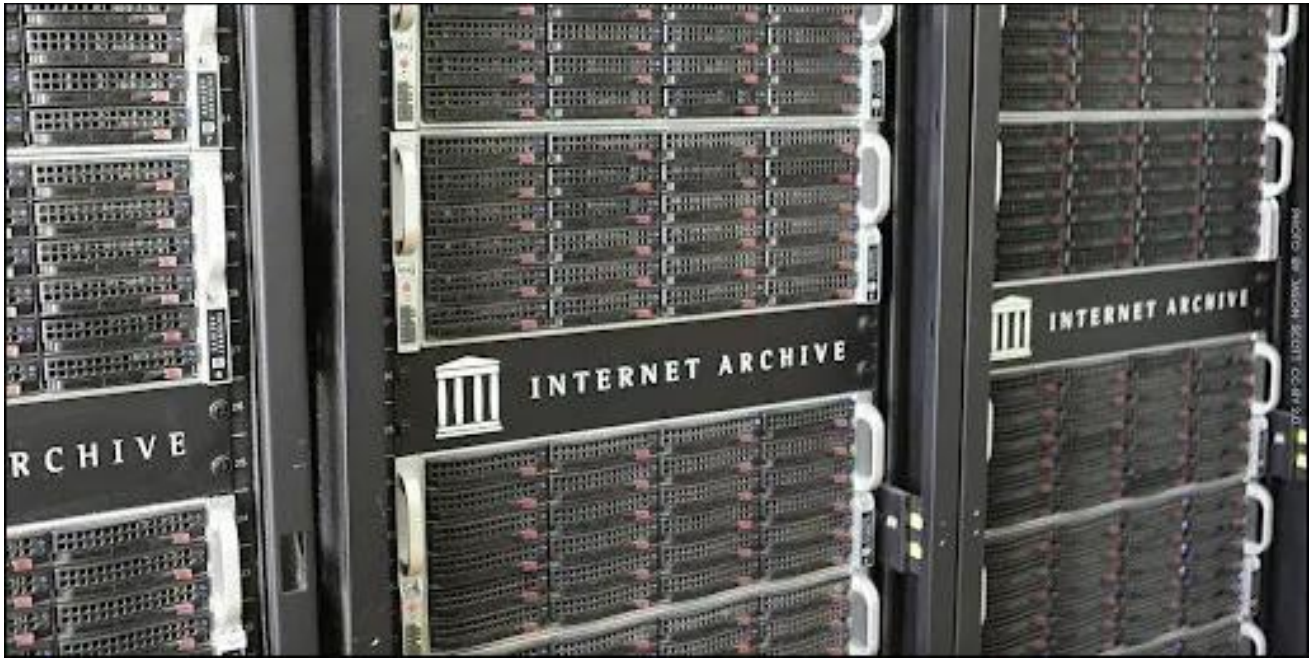
commandlinefu.com

Screenshot Showcase

Posted by mutse, on April 15, 2026, running mate.

Blocking The Internet Archive Won't Stop AI, But It Will Erase The Web's Historical Record

by Joe Mullin
Electronic Frontier Foundation
Reprinted under Creative Commons license



Imagine a newspaper publisher announcing it will no longer allow libraries to keep copies of its paper.

That's effectively what's begun happening online in the last few months. The Internet Archive—the world's largest digital library—has preserved newspapers since it went online in the mid-1990s. The Archive's mission is to preserve the web and make it accessible to the public. To that end, the organization operates the Wayback Machine, which now contains more than one trillion archived web pages and is used daily by journalists, researchers, and courts.

But in recent months The New York Times began blocking the Archive from crawling its website, using technical measures that go beyond the web's traditional robots.txt rules. That risks cutting off a record that historians and journalists have relied on for decades. Other newspapers, including The Guardian, seem to be following suit.

For nearly three decades, historians, journalists, and the public have relied on the Internet Archive to preserve news sites as they appeared online. Those archived pages are often the only reliable record of how stories were originally published. In many cases, articles get edited, changed, or removed—sometimes openly, sometimes not. The Internet Archive often becomes the only source for seeing those changes. When major publishers block the Archive's crawlers, that historical record starts to disappear.

The Times says the move is driven by concerns about AI companies. There's a strong case that such training is fair use.

Whatever the outcome of those lawsuits, blocking nonprofit archivists is the wrong response. Organizations like the Internet Archive are not building commercial AI systems. They are preserving a record of our history. Turning off that preservation in an effort to control AI access could essentially torch decades of historical documentation over a fight that libraries like the Archive didn't start, and didn't ask for.

If publishers shut the Archive out, they aren't just limiting bots. They're erasing the historical record.

Archiving and Search Are Legal

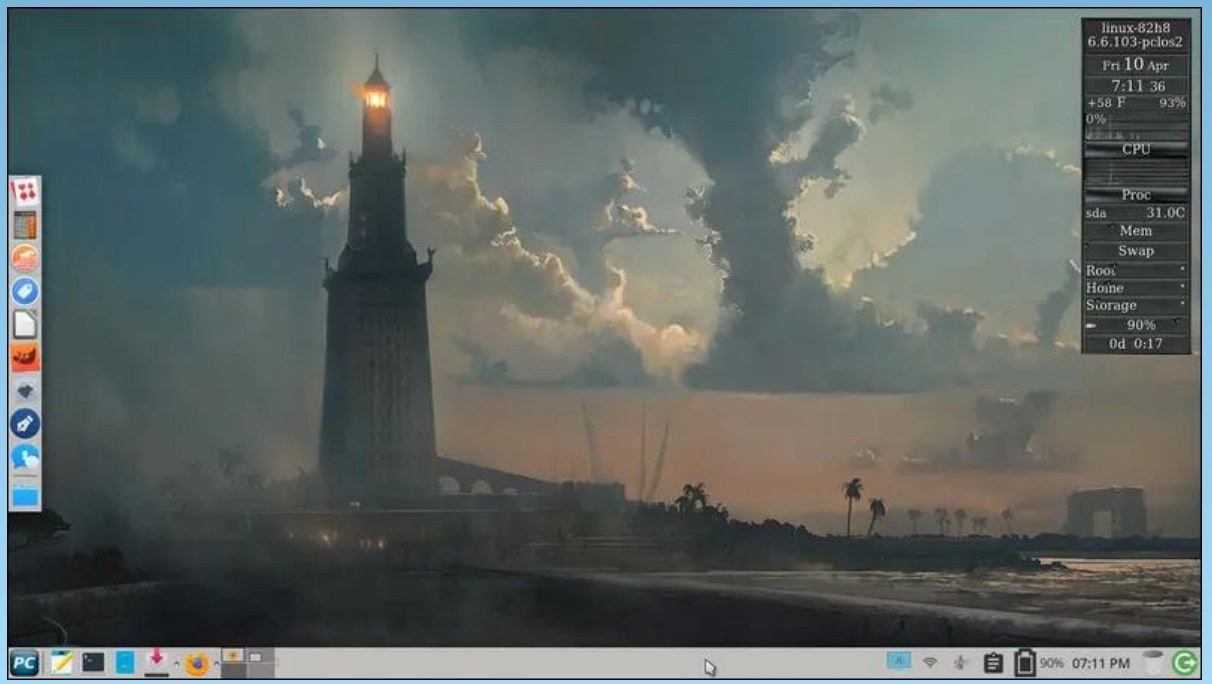
Making material searchable is a well-established fair use. Courts have long recognized it's often impossible to build a searchable index without making copies of the underlying material. That's why when Google copied entire books in order to make a searchable database, courts rightly recognized it as a clear fair use. The copying served a transformative purpose: enabling discovery, research, and new insights about creative works.

The Internet Archive operates on the same principle. Just as physical libraries preserve newspapers for future readers, the Archive preserves the web's historical record. Researchers and journalists rely on it every day. According to Archive staff, Wikipedia alone links to more than 2.6 million news articles preserved at the Archive, spanning 249 languages. And that's only one example. Countless bloggers, researchers, and reporters depend on the Archive as a stable, authoritative record of what was published online.

The same legal principles that protect search engines must also protect archives and libraries. Even if courts place limits on AI training, the law protecting search and web archiving is already well established.

The Internet Archive has preserved the web's historical record for nearly thirty years. If major publishers begin blocking that mission, future researchers may find that huge portions of that historical record have simply vanished. There are real disputes over AI training that must be resolved in courts. But sacrificing the public record to fight those battles would be a profound, and possibly irreversible, mistake.

Screenshot Showcase



Posted by Meemaw, on April 10, 2026, running Xfce.

PCLinuxOS Recipe Corner Bonus

Low Country Shrimp

Serves: 4

INGREDIENTS:

1 lb small red potatoes, halved
4 pieces frozen mini corn on the cob, thawed, cut in half
2 teaspoons oil
2 teaspoons Old Bay™ seasoning
1 lb uncooked peeled deveined extra-large shrimp (16 to 20 count)
12 oz fully cooked andouille sausage, sliced
1 lemon, cut into 8 wedges
¼ cup chopped fresh parsley leaves

DIRECTIONS:

1. Heat gas or charcoal grill. Cut 4 (18x12-inch) sheets of heavy-duty foil. Spray with cooking spray.

2. Place potatoes in a microwavable bowl. Microwave uncovered on High 5 to 6 minutes or until potatoes are just tender. Add corn to potatoes; drizzle with 1 teaspoon of the oil, and sprinkle with 1 teaspoon of the seasoning; mix until evenly coated. Place shrimp in a medium bowl; toss with remaining 1 teaspoon oil and remaining 1 teaspoon seasoning; mix until evenly coated.

3. Place an equal amount of sausage on the center of each sheet of foil. Dividing evenly, arrange potato and corn mixture around sausage. Divide shrimp evenly over sausage. Squeeze 1 wedge of lemon over each pack.

4. Bring up 2 sides of foil so edges meet. Seal edges, making tight 1/2-inch fold; fold again, allowing space for heat circulation and expansion. Fold other sides to seal.

5. Place packs on the grill over medium heat. Cover grill; cook for 6 minutes. Rotate packs 1/2 turn; cook 5 to 7 minutes longer or until shrimp are pink and sausage is heated through. Remove packs from the grill, cut large X across the top of each pack. Carefully fold back foil; sprinkle with parsley, and top with remaining lemon wedges.

Options:

Two ears of fresh sweet corn can be substituted for the frozen corn in this recipe. Cut each ear into 4 pieces.

To make it in the oven, place the packs on a cookie sheet. Bake at 375°F 23 to 25 minutes or until shrimp are pink and sausage is heated through.

Inspiration & Motivation

PCLinuxOS Puzzled Partitions

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

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 226, average score 158.

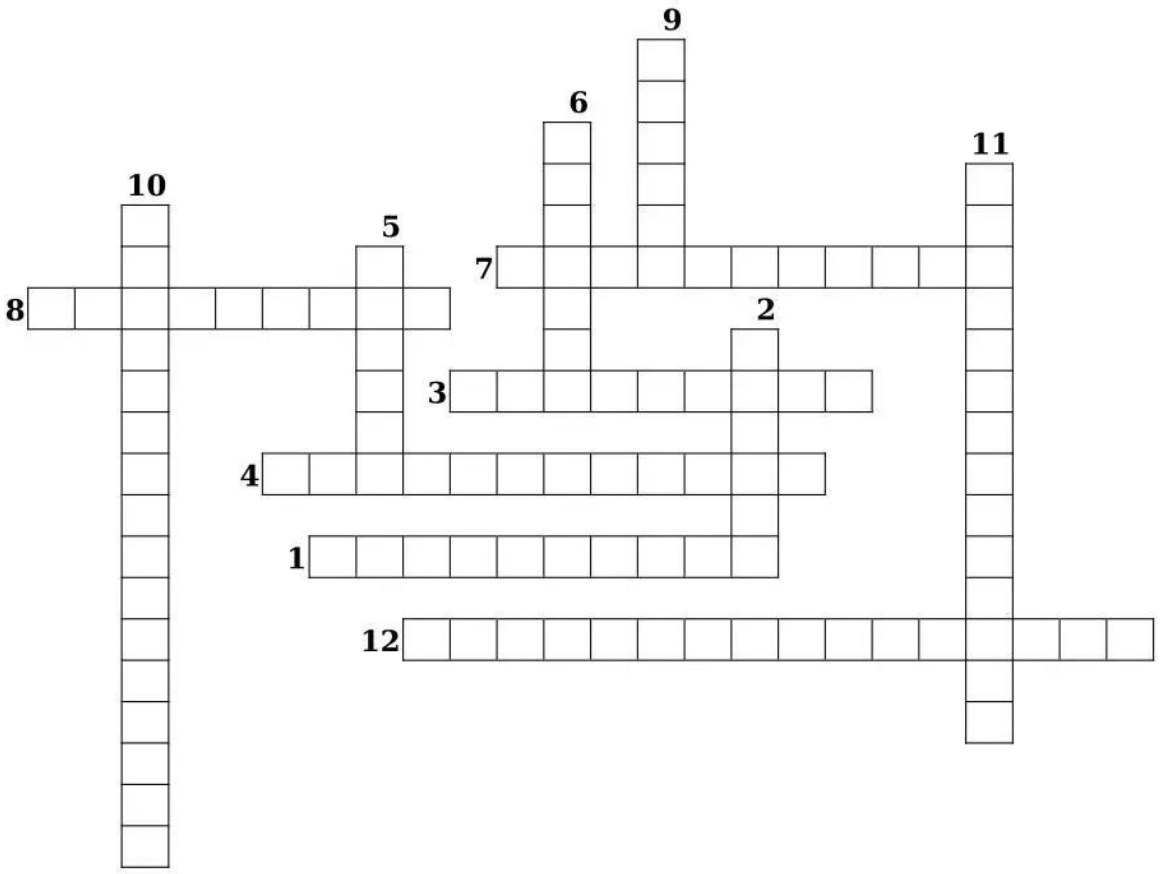
May 2026 Word Find
Space!

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

AERONAUTICS
APOLLO
ARTEMIS
ASTRONAUT
COSMOS
COSMONAUT
DOCKING
EAGLE
ENGINEERS
ENGINES
EXTRATERRESTRIAL
FREEDOM
FRIENDSHIP
GALAXY
GEMINI
GRAVITY
HEAT SHIELDS
INTERSTELLAR
LAUNCH
LUNAR MODULE
MERCURY
MISSION CONTROL
NAVAL AVIATOR
ORBIT
OUTER SPACE
PLANETS
ROCKET
ROCKET SCIENTIST
SATELLITE
SCIENTISTS
SEA OF TRANQUILITY
SOLAR SYSTEM
SPACE EXPLORATION
SPACEWALK
TECHNOLOGY

Download Puzzle Solutions Here

May 2026 Crossword
Space!



1. The application of science, especially to industrial or commercial objectives.
2. A system of stars and other celestial objects bound by gravity.
3. An astronaut of the Soviet or Russian space program.
4. Occurring between or among the stars.
5. A large, rounded astronomical body that orbits a star, such as the Sun.
6. The current mission to send a crew of four astronauts on a 10-day journey around the moon.
7. The science of designing, building, and operating aircraft.
8. Any activity by an astronaut outside of a spacecraft or space station.
9. A program of space flights undertaken by US to land a man on the Moon.
10. A vast, dark plain on the Moons surface.
11. The group of people on the ground who direct or control the flight of a spacecraft.
12. Originating from outside of the Earth's atmosphere, from space, or from another planet.

Download Puzzle Solutions Here

Mixed-Up-Meme Scrambler

LEJUP

GADEA

KURBEE

NULRUY

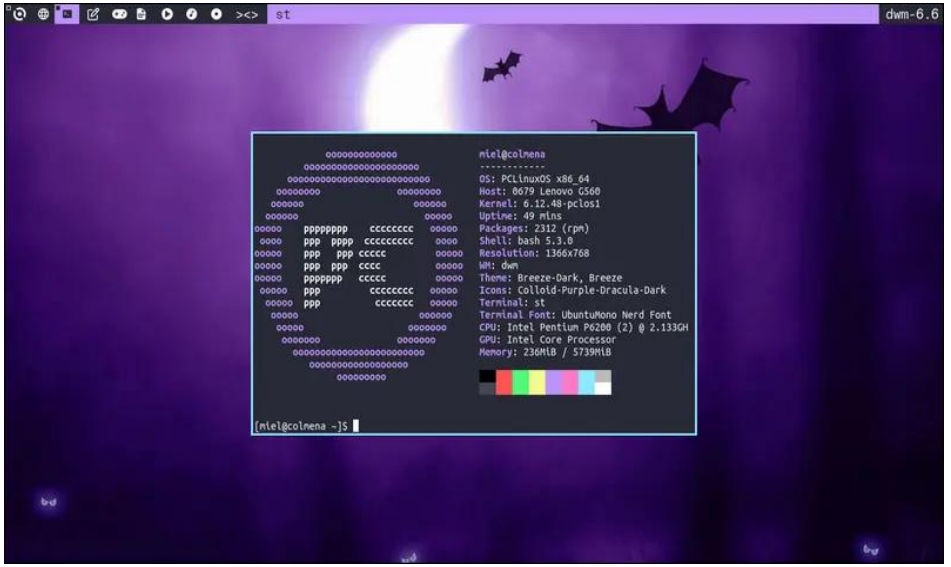
Good at finding things before they are missing ...

A "_ _ _ _ _ _ _ _"

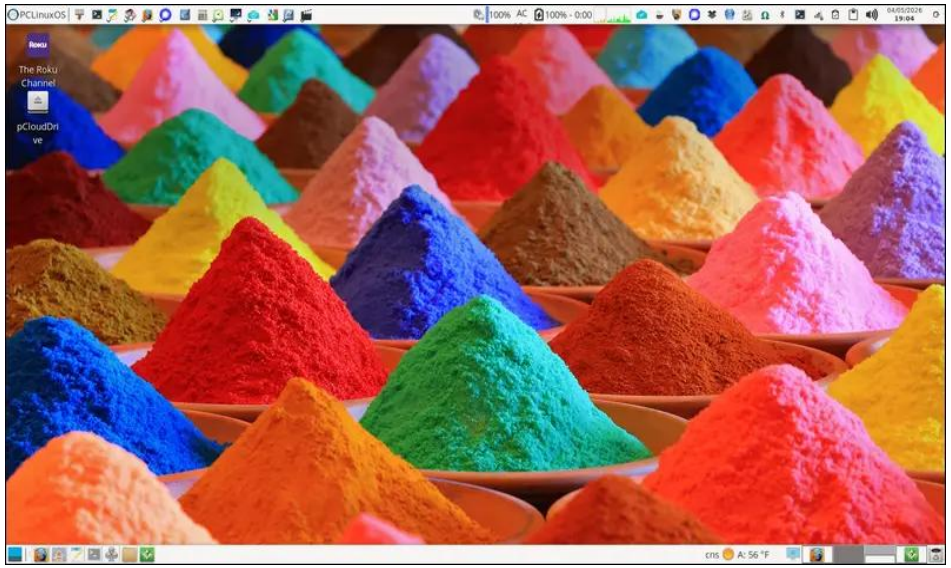
Download Puzzle Solutions Here

More Screenshot Showcase

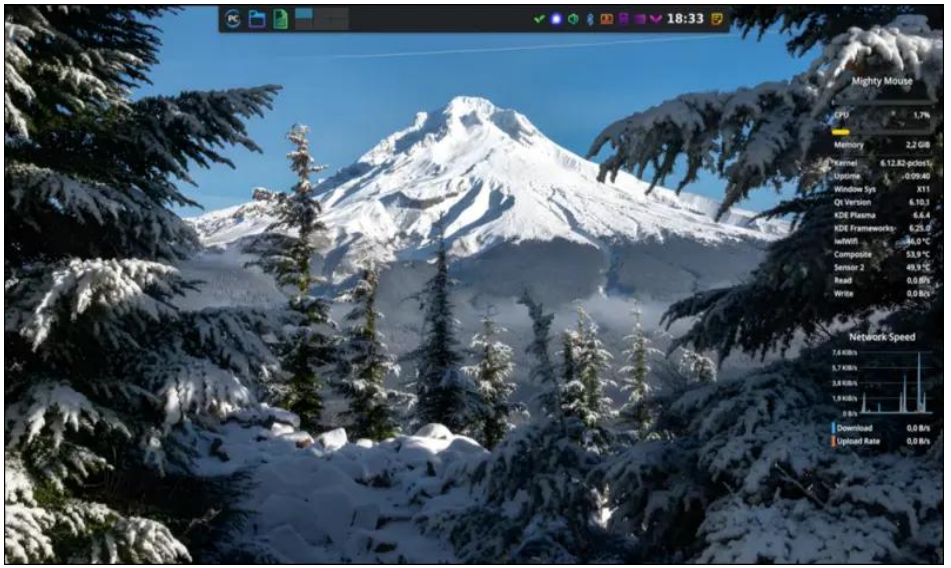
Posted by mutse, on April 3, 2026, running KDE.



Posted by kimiko, on April 8, 2026, running DWM.



Posted by parnote, on April 5, 2026, running Xfce.



Posted by tbs, on April 22, 2026, running KDE.