The PCLinuxOS magazine
Volume 147
April, 2019

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 http://www.pclosmag.com

This release was made possible by the following volunteers:

Chief Editor: Paul Arnote (parnote)
Assistant Editor: Meemaw
Artwork: ms_meme, Meemaw
Magazine Layout: Paul Arnote, Meemaw, ms_meme
HTML Layout: YouCanToo

Staff:
ms_meme
Meemaw
Gary L. Ratliff, Sr.
Daniel Meiß-Wilhelm
daiashi
Cg_Boy
YouCanToo
Pete Kelly
Smileeb
Alessandro Ebersol

Contributors:
BillDjr

From The Chief Editor's Desk ...

I don't know about you, but for me, there's little as exciting as installing PCLinuxOS on a "new" computer. Now, "new" doesn't have to be "brand new." It just has to be new to me.

Ever since PCLinuxOS dumped the 32 bit versions, I've felt like a stranded sailor on a deserted island. I used to carry around a netbook (remember those?) that I would use when I was at work, or out and about. However, that netbook only had a 32 bit Intel Atom processor. That meant that I could no longer update it, beyond the last iteration of the 32 bit PCLinuxOS repository – which I did right before the 32 bit repository shut down.

I never minded running the 32 bit version of PCLinuxOS, even on my computers with a 64 bit processor. Since I always used the PAE kernel, the 32 bit kernel's memory restriction was never an issue. It just ran, and ran dependably. Plus, I wasn't too keen on seeing a perfectly good computer go into the trash.

I kept that netbook going for as long as I could, limping along without updates. Then, finally, that netbook literally died a few months back. It wouldn't power on any longer. I had been looking for a replacement, at least looking casually. For a while, I carried a full size laptop, but the weight was atrocious. Something had to give. That started me, at last, on a quest to replace that netbook. Meanwhile, I salvaged all I could from the dead netbook … RAM, hard drive, etc. Anything that might be useful was saved, while the rest of the computer took up permanent residence in the local landfill.

It took me purchasing two used laptops to find a suitable replacement. It had to have a small form-factor. The first one, a Lenovo x131e was a laptop targeted at the educational market. It had one of

those AMD "E" series processors, 4GB RAM and a 320GB hard drive. But, the video processor was anemic and slow. I couldn't even stream YouTube videos without extreme buffering, despite a more-than-adequate network speed. But then, I only paid $64 (U.S.) for it on eBay, so I wasn't out a whole lot of money. I kept looking.

The second laptop I purchased was actually the one I found that I preferred, even before I purchased the Lenovo x131e. I had been looking at them – a lot – when I purchased the first one. Since I already had the x131e, I wasn't in any hurry. I figured I'd take my time and look for just the right deal. It is a Lenovo x230, with an Intel i5 processor, 8GB RAM, and a 128GB SSD (yes, I like Lenovo laptops … especially the ThinkPad line). When this laptop was new, it originally sold for over $1,250. I found several that were missing the hard drive, and/or the hard drive caddy. Many only had 4GB RAM. Some would require a bit of work to make functional, and I was not opposed to putting in a little work on it, just so long as the price was right. But this one almost seemed too good to be true, and you know what is typically said about THAT situation. For less than $150, I decided to "pull the trigger" and took a chance.

I can't say in the least that I was sorry for taking that chance. When I received that laptop, I was astonished. This laptop was in such excellent shape. It looked like it had hardly been used at all. I was originally fearful that the seller had just used "stock" images of the laptop, from when the laptop was new. That fear proved to be unfounded.

Is it fast, you ask? Yep! That 128GB SSD makes PCLinuxOS FLY! This isn't my first experience with a SSD. My desktop system that I built a while back also has a SSD.

Is the laptop everything I wanted? Not exactly. I would have preferred to have a larger SSD. But, I

can live with this one. To bolster my storage space, Newegg had a 128GB microSD card on sale a couple of weeks ago, complete with an adapter to allow it to be used in a standard SD card slot. Score! Portable storage, and only for less than $17!

So, I got to install PCLinuxOS on TWO "new" computers. In both instances, PCLinuxOS installed easily (from a LiveUSB). All hardware was properly recognized and setup. In the case of both laptops, each was fully installed and ready to go in less than 30 minutes.

Like I said before, there's nothing quite like installing PCLinuxOS on a "new" computer. On the x131e, PCLinuxOS replaced Windows 7. On the x230, PCLinuxOS replaced Windows 10. I only ever booted into each one to make sure that everything worked. After that, I wiped the hard drives of each laptop, set up my custom disk partitions, and installed PCLinuxOS.

What joy!

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

PCLinuxOS Radically Simple

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.

Casual Python, Part 3

by Peter Kelly (critter)

Stuff

One of the many books that I read while trying to learn python was 'Learning Python,' by Mark Lutz. In the book, he wrote "Programs do things with stuff." That just about sums it up and brings it down to our level. So, what "stuff" and what "things"? The complete answer turns out to be "Just about anything with anything," but for now we shall have to be content with just doing normal things to the easy to understand stuff.

Python calls the 'stuff' that he talks about 'objects,' and python has a lot of objects. In fact, in python, everything is an object. To do the things it does with these objects, python generally uses functions or methods. There are many different types of objects, and python has a few 'core' types about which we really need to know if we are going any further. Each of the types has its own set of methods, and these we also need to be familiar with.

The core types can be listed as:
- Numbers
- Strings
- Lists
- Tuples
- Dictionaries
- Sets
- Files

There are possibly more, but there is a bit of an overlap between objects, such as numbers and booleans, so the distinction is blurred. Our next project uses strings, lists and files, so I will describe the basics of those. Numbers are something we have already covered.

Strings

A string is an ordered sequence of characters. The characters may be alphanumeric, punctuation, whitespace, or anything defined by the utf-8 standard. Don't worry too much about the last bit. If you can type it, then it is covered by utf-8, which is how the computer knows how to display it. All programming languages use strings. Some, such a C, also have a char type which holds a single character. Python does not. In python, that is a string of length one. Strings in python can be any length that will fit in available memory (but if you really need a really huge string, consider using a file).

A string in python is an object of type str and some other objects may be converted to strings by using the str() function.

```
>>> str(11.75)
'11.75'          # The float type has been converted to a string type.
```

Strings are 'immutable,' which means resistant to mutation, i.e., they cannot be changed. That may seem to be a big disadvantage, but in fact, as you will see, it is just the opposite, and there are other ways to get new strings.

A string is usually referenced by assigning it to a variable, which really means giving it a name.

```
s = 'The python language'
```

Any time python is passed the variable s, it is immediately replaced by the text it represents. Since strings are immutable, you can be certain of what that will be. No other part of the program can have changed it. Strings can be chopped up by a process known as slicing, but the original is unchanged. Any slices that need to kept will have to be given a new name (assigned to a variable). Slicing is done using square brackets containing a start position, stop position and optional step or 'stride'. The stop character is not included. If start or stop is omitted, the the beginning or end is assumed. The characters are indexed starting with 0, so that we have:

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
T h e   p y t h o n    l  a  n  g  u  a  g  e
18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
```

```
s[0:6] ==> 'The py'
s[4:10] ==> 'python'
s[:10] ==> 'The python'
s[:10:2] ==> 'Tepto'      # from the beginning, up to but not including pos'n 10 with step 2
s[:] ==> 'The python language'
```

The last is a quick way to make a copy.

Negative values count backwards from 1 (there can be no minus zero):

```
s[-3:] ==> ‘age’
s[-12:-4] ==> ‘hon lang’
s[:-9] ==> ‘The python’
```

s still references ‘The python language.’ That cannot be changed. This takes practice, but is invaluable, so persevere.

```
t = s
```

Both s and t reference the same string, ‘The python language’

The way ‘things’ are done to strings is by methods, and there is a method for just about anything you would want to do to a string. Methods are called using ‘dot notation’

```
s.upper() ==> 'THE PYTHON LANGUAGE'
s.find('python') ==> 4, the start index of the substring ‘python’
s.replace('python', 'English') ==> 'The English language'
s.split() returns a list of strings ==> ['The', 'python',
'language']
```

There are a lot of these methods, which are documented in the official documentation https://docs.python.org/3/, but you can get a good start by reading C H Swaroops book that I mentioned earlier: https://python.swaroopch.com/.

When I covered numbers, I showed how to use python’s built-in interpreter. However, there is a better version, which is available for installation from the repositories, named ipython. This is a straight replacement for the standard interpreter, but has many more features. The first difference that you will notice is that the prompt has changed from >>> to '`In [1]:`' and is in color, green by default. When you enter an expression and press return, a new line is output beginning with '`Out [1]:`' in red by default, which is followed by the output from the expression. Each time you enter a new expression, the number in the brackets is incremented. This is your command history. There are a whole lot more benefits to using this interpreter, some of which we will cover shortly.

Doing things with stuff is more generally referred to as 'data processing' and, when the stuff is text, 'text processing.'But, as the venerable Mr. Lutz pointed out, we are still only doing things with stuff. Although the string cannot be changed, the object that the variable s references can be, and it can be changed to any object type.

```
s = s[4:10] ==> 'python'
```

Now s references the new string ‘python,’ which was obtained by slicing. The original string is untouched so that any other variable names that referenced it, such as t, will not be inconvenienced. If no other references to it have been made, then it will be automatically removed from memory by a process known as ‘garbage collection’.

```
s = 3.14159
s now references a floating point number ( decimal).
t ==> ‘The python language’ .
```
t is unchanged

Strings support concatenation and repetition by adding and multiplying.

```
t + ‘ is powerful’ ==> 'The python language is powerful'
t[4:10] * 3 ==> 'pythonpythonpython'
```

This can be combined

```
(t[4:10] + ‘ ‘) * 3 ==> 'python python python '
```
spaces added.

The characters in strings have to be quoted using either single or double quotes. Which you use doesn’t matter, but if a string contains a quote mark, then the other type should be used.

```
‘He said “Hello” to the stranger’

“Pythons’ syntax”
```

There are a third type of quotes in python, called triple quotes. These can be single or double, and allow strings to span multiple lines. They are commonly used in code documentation.

```
””” This function calculates the square of a number.
sq(x)
x is the number to be squared
returns the square of x """
```

Using the interpreter, here I am using the ipython interpreter, type the following:

```
In [2]: dir(str)
Out [2]:
['__add__',
'__class__',
'__contains__',
...
... many more methods deleted
...
'rstrip',
```

```
'split',
'splitlines',
'startswith',
'strip',
'swapcase',
'title',
'translate',
'upper',
'Zfill']
```

This gives a full list of the methods available for use with strings. You can ignore the ones that start with a double underscore for now. If you type, for example:

In [3]: help(str.s

Press the tab key and you will be prompted with the names of the methods that start with s

```
In [3]: help(str.s
str.split str.strip
str.splitlines str.swapcase
Str.startswith
```

Use the arrow keys to select one, close the parentheses and press return

```
In [3]: help(str.strip)
```

Help on method_descriptor:

```
strip(self, chars=None, /)
```

Return a copy of the string with leading and trailing whitespace remove.

If chars is given and not None, remove characters in chars instead. This is some basic help on the method. Press 'q' to return. Try it out.

```
In [4]: s = ' the end '

In [5]: s.strip()
Out[5]: 'the end'
```

This tab completion is a great help when typing expressions. The color coding extends to different object types, which is also helpful. Your history, but not the values in variables, is saved across sessions. Press the up arrow to go back.

Lists

In python, a list is an ordered collection of ‘things’, which are listed between square brackets and separated by commas. The ‘things’ can be any type of object, even other lists. Lists, unlike strings, are mutable, they **can** be changed in place.

```
list1 = ['The', 'python', 'programming', 'language'] # a list of
strings
list2 = [42, 3.14159]         # a list of two types of number
list3 = ['Linux', 17, 42]     # a mixed list
list4 = []                    # an empty list
list5 = list1             # both list1 and list5 reference the same
# list
list1 == list5 ==> True       # has the same value
list1 is list5 ==> True       # references the same object
```

A single '=' is used for assignment, '==' is used to test for equality.

Lists are sliceable like strings.

```
list6 = list1[:]              # list6 is a copy of list1
list6 == list1 ==> True       # has the same value
list6 is list1 ==> False      # reference different objects
```

Also like with strings, concatenation and repetition are supported. The result of these operations are a single new list, which must be assigned to a variable to be kept. The original list(s) are not changed. The multiplier in repetition must be an integer.

```
list1 = [7, 3]

In [6]: list1 + list1
Out[6]: [7, 3, 7, 3]

In [7]: list1 * 3
Out[7]: [7, 3, 7, 3, 7, 3]

In [8]: list1
Out[8]: [7, 3]

In [9]: list1 * list1
---------------------------------------------------------------------
---------
TypeError Traceback (most recent call last)
<ipython-input-9-86e343e4b412> in <module>
----> 1 list1 * list1
```

```
TypeError: can't multiply sequence by non-int of type 'list'
```

Lists are also indexable.

```
list1[1] ==> ‘python’
list3[2] ==> 42
```

Elements can be 'popped' # removed from the end.

```
list1.pop(2) ==>  'programming'
list1.pop()  ==>  'language'
list1 is now      [‘The’, ‘python’]
```

They can also be cleared.

```
list3.clear() ==> []
```

Elements can be appended.

```
list2.append(1.6e3) ==> [42, 3.14159, 1600.0]
                  # 1.6e3 is 1.6 * 10³
```

Or inserted at a position before an index.

```
list2.insert(2, 4096) ===> [42, 3.14159, 4096, 1600.0]
```

And they can be sorted.

```
list2.sort() ==> [3.14159, 42, 1600.0, 4096]
list1.sort() ==> ['The', 'language', 'programming', 'python']
```

As strings are mutable, this sort is carried over to all variables that reference it.

```
list5 ==> ['The', 'language', 'programming', 'python']
```

Immutable objects are not such a bad thing.

The sort can be controlled by use of a key.

```
list6 ==> ['The', 'python', 'programming', 'language']
list6.sort(key=str.lower)    ==>    ['language',    'programming',
'python', 'The']
```

Here strings lower method, which returns a lowercase version of the string, is used as the sort key.

This uses the list objects own sort method, but there is also a built in function named sorted (functions like this are known unsurprisingly as ‘builtins’). This can be used when the object type has no sort method of its own. The sorted object is not automatically updated, and must be manually assigned to a variable name. The items to be sorted must be of the same type, or you will get an error.

```
list5 = [99, 17, 42]
list5 = sorted(list5) ==> [17, 42, 99]
```

However,

```
list6 = [17, 'Linux', 9]
list6 = sorted(list6)
Traceback (most recent call last):
File "list_sort.py", line 13, in <module>
list6 = sorted(list6)
TypeError: '<' not supported between instances of 'str' and 'int'
```

You will get a lot of errors like this when you first start writing your own code. As we get more complex, I’ll show how to find and deal with errors – something even very experienced programmers have to do, so don’t be discouraged.

Files

Files are the permanent storage of the 'stuff' we are doing 'things' to. Usually, the storage is a hard drive partition on the local or a remote computer. Remotely stored files are typically on a networked computer, web based or on cloud storage systems but to python there is little difference.

To use a file you have to know its location, and have access rights appropriate for reading, writing, creating or deleting the file or directory. The file has to be opened in the correct mode for the action you wish to perform, and there are several modes available:

‘r’ is read mode, the default if not stated.

‘w’ is write.

‘a’ is append.

'b' is for binary or byte files.

't' is for text files, the default, but if omitted is assumed, so is rarely seen. 'wt' means write in text mode'

Adding a '+' e.g. 'r+' provides read and write mode, and if the mode is 'w' or 'a', then the file is created if it does not exist.

Caution! Write will overwrite whatever is already there. If that is not what you want, then use append. Without the 'b', files are opened in text mode. This is what we mostly want, but if you are working with binary data such as graphics or audio, the the file must be opened in bytes mode. The two are not interchangeable, and using the wrong mode will result in garbage at best, and data loss at worst.

For now, we'll stay with text mode. When you have finished with the file, it must be closed. Usually, python will close any open files for you when the program terminates. But if the program crashes while the file is still open, there is a very real chance of data loss. To make our lives easier, python has something called a context manager that ensures that files are closed, even in the event of a crash.

If we do something like:

```
with open('myfile.txt', mode='a+') as f:
f.write( 'Adding to file...\n')
```

Write does not automatically add an end of line (\n) so we have to do it. Then run this code 5 times, if we then run *this* code:

```
with open('myfile.txt', mode='r') as f:
print(f.read())
```

We will see:

```
Adding to file...
Adding to file...
Adding to file...
Adding to file...
Adding to file...
```

Alternatively we could use the file objects readline method to read a single line.

```
with open('myfile.txt', mode='r') as f:
print(f.readline(), end='')

adding to file...
```

The end='' in the print statement is necessary, as the print function **does** add a newline character by default (to the one we added when writing the file), as we can see if we iterate over the file:

```
with open('myfile.txt', mode='r') as f:
    for line in f:
        print(line)

adding to file...

adding to file...

adding to file...

adding to file...

adding to file...
```

This may seem strange, but when we write to a file, we want to control exactly what gets sent to the file. When we print out something, we usually want a newline, but can override this when necessary with the end= clause. When we used print(f.read()) earlier, the print function output what was in the file, and then added a final newline.

But what if….?

If a script could only execute one line at a time in the order they appeared, we would be severely restricted. Sometimes, decisions have to be made and, depending on the result of that decision, a different action taken. Occasionally, the same 'things' need to be done to a whole bunch of 'stuff'. This is called looping and branching, or program control.

Decision making can be controlled with the **if** statement and the optional **elif** and **else** statements. This takes the form

if condition is true:
  do things to stuff

elif a different condition is true:
  do other things to stuff

Else:
  do this

There is also a **while** statement that looks like this:

while condition is true:
  do things to stuff

This will continue doing things to stuff until the condition is no longer true.

There is also a **break** statement that jumps straight out of the loop, and a **continue** statement that cancels just the current iteration of the loop and continues with the next.

To do things to groups of stuff, we have the **for** statement that was used to iterate over the file in the example above.

for all of stuff in this collection of stuff:
    do these things

Keep an eye out for these in the code of the next few applications, which will start to use these new structures.

With either the standard interpreter or ipython, there is a continuation prompt for multiline expressions such a loops which is three dots:

```
In [6]: for i in s.strip():
...: if i == ' ':
...: print('Found a space')
...:

Found a space
```

That's enough theory for now.

Docz revisited

So far, we haven't used a lot of python code, because we don't really know much. This is about to change, as in the next application, we put to use some of the theory just covered. This application doesn't just use a command line tool to perform a function. It does some real file handling and text processing in a mouse and keyboard sensitive GUI. And it is actually quite useful.

We want to make this application do more than just display the names of some files. To be useful, it should launch the associated application so that we can work on it. In this case, the application to be launched is LibreOffice Writer, and the command to do this is

```
Libreoffice --writer filename.odt
```

Where filename.odt is supplied by the docz application.

To make things easier, we will change to the directory where the files are stored. To do this, we need to import another standard module named os, which gives us some useful operating system commands. The methods we are looking for are os.path.expanduser() and os.chdir(). The first expands the string '~' to the path of the user's home directory, the second changes directory. We shall also have to tell the application what to do when the mouse is clicked on a filename (and what to do if it is clicked in the empty space after the list of files).

For the application to know where in the text we clicked, we need a cursor, which Qt can provide. However, we need to add another Qt import statement, as it is not provided by either of the two imports we already have. This is fairly advanced stuff, but we can use it even if we don't really understand it.
The modified code looks like this.

```
#!/usr/bin/env python3

import sys
from PyQt5.QtCore import *
from PyQt5.QtGui import *
from PyQt5.QtWidgets import *
import subprocess
import os
import docz_ui

class Docz(QWidget, docz_ui.Ui_Form):
    def __init__(self):
        super(self.__class__, self).__init__()
        self.setupUi(self)
        self.quitButton.clicked.connect(self.exitApplication)
        home = os.path.expanduser('~')
        os.chdir(home + '/Documents/wills_plays/')
        cmd = "ls -1 *.odt" # that's -one not -ell
        result = subprocess.check_output(cmd,
shell=True).decode('utf-8')
        self.resultslist.setText(result)
        self.resultslist.mousePressEvent = self.mousePressed

    def mousePressed(self, Event):
        textCursor =
self.resultslist.cursorForPosition(Event.pos())
        textCursor.select(QTextCursor.BlockUnderCursor)
        self.resultslist.setTextCursor(textCursor)
        f_name = textCursor.selectedText()
        if f_name == '': # mouse was clicked in empty space
            pass         # so ignore it
        else:
            f_name = f_name[1:] # drop the first character
            command = ('libreoffice --writer ' + f_name)
            subprocess.Popen(command, shell=True)
            self.exitApplication()
```

```
    def keyPressEvent(self, e):
        if e.key() == Qt.Key_Escape:
            self.exitApplication()

    def exitApplication(self):
        self.close()
        sys.exit()

if __name__ == '__main__':
app = QApplication(sys.argv)
form = Docz()
form.show()
app.exec_()
```

If you run the code as supplied, make sure that you have some writer files in your home directory for the application to find.

So, what have we done? We added a line to the initialization code that connects a mouse press to a new mousePressed method. The first four lines in this method set up a QtextCursor, which is how Qt determines its current position in the text. Also, in this new method, we use an if - else routine to check if the mouse was clicked on some text or in empty space. The first character of the returned filename is an invisible character, known as a paragraph separator, which we don't want. So, we assign the variable f_name to a slice of the string that does not contain it. This is how we chop strings up and re-assign them, as strings cannot be changed; they are immutable.

if the text was blank - pass. Pass is python's do-nothing command which does - nothing!

else build a command using the text referred to by word.

execute the command and shut down the application.

Finally, open designer and change the textedit font size to 12 or 14. This will make it easier to select the correct file. Save it and run update_res.sh.

One thing to note here is that although we have a *keyPressedEvent* method in our code, we are using the Qt textedits' *mousePressEvent* method, so we should name our mouse click detection routine differently to avoid possible conflicts hence *mousePressed*.

International Community PCLinuxOS Sites

PCLinuxOS.nl
Netherlands

PCLinuxOS
Daykonun mekanı
Turkey

PC Linux OS
Denmark

Czechoslovakia
PCLinuxOS
síla jednoduchosti

PCLinuxOS.it
Italy

PCLinuxOS Polskie Centrum
Poland

PCLinuxOS FINLAND

PCLinuxOS
Brazil

Screenshot Showcase

Posted by daniel, March 4, 2019, LXQt.

PCLinuxOS Recipe Corner

Skillet Chicken with Roasted Garlic and Mushroom Gravy

Ingredients

2 garlic bulbs
2 teaspoons olive oil
4 boneless skinless chicken breasts (about 1 1/4 lb)
1/2 cup all-purpose flour
1/2 teaspoon dried thyme leaves
1/2 teaspoon salt
1/4 teaspoon pepper
1/4 cup butter
8 oz sliced mushrooms
1 1/2 cups chicken stock (from 32-oz carton)
1/2 cup heavy whipping cream

Directions

1. Heat oven to 350°F.

2. Cut 1/4 to 1/2 inch from top of each garlic bulb to expose cloves. Place each cut side up on 12-inch square of foil. Drizzle each bulb with 1 teaspoon of the oil. Wrap securely in foil. Place in pie plate or shallow baking pan. Bake 45 to 50 minutes or until garlic is tender when pierced with toothpick or fork. Cool slightly. From root end, squeeze soft cloves out of papery skins, and chop finely. Set aside.

3. Meanwhile, between pieces of plastic wrap or waxed paper, place each chicken breast smooth side down; gently pound with flat side of meat mallet or rolling pin until about 1/4 inch thick. In shallow pan, stir together flour, thyme, salt and pepper. Reserve 3 tablespoons of the seasoned flour for the sauce. Coat both sides of chicken with remaining flour mixture.

4. In 12-inch skillet, heat 2 tablespoons of the butter over medium-high heat. Cook chicken in butter 6 to 8 minutes, turning once, until no longer pink in center (at least 165F). Transfer chicken to plate; cover to keep warm.

5. In same skillet, heat remaining 2 tablespoons butter over medium heat. Cook mushrooms in butter until lightly browned. Sprinkle the 3 tablespoons reserved seasoned flour over mushrooms. Cook and stir 2 minutes. Stir in chicken stock, cream and roasted garlic. Heat to boiling. Reduce heat to low; simmer about 3 minutes or until slightly thickened.

6. Place chicken back into skillet with sauce about 1 minute or until heated through. Serve with mashed potatoes, if desired.

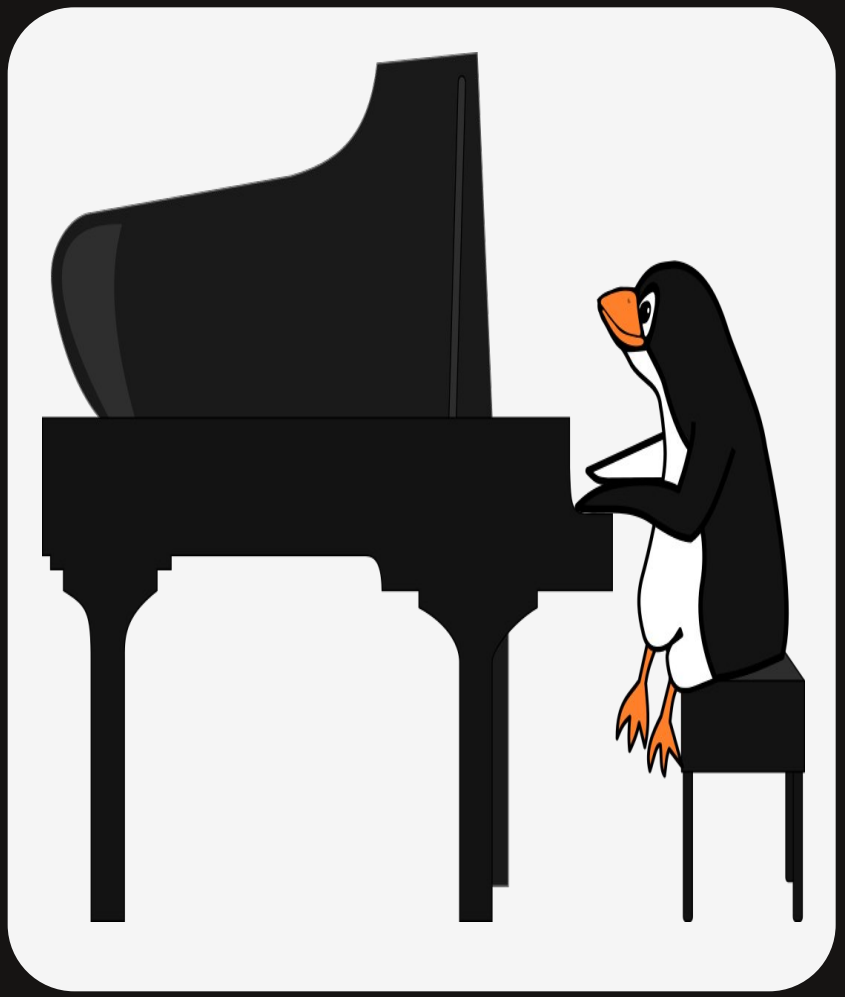
Tips:

For a walk on the wild side, try using sliced cremini or shiitake mushrooms in this recipe.

If you prefer to use fresh herbs instead of dried, here is a good rule of thumb to follow. Fresh herbs are generally less potent and concentrated than dried herbs, so you'll need more -- typically three times the amount of fresh herbs as dry. For instance, this recipe calls for 1/2 teaspoon dried thyme, so you would use 1 1/2 teaspoons finely chopped fresh thyme.

commandlinefu.com

ms_meme's Nook: The PCLOS Rag



MP3 OGG

A new OS is sweeping the net
Made by Texstar no sweat
Download put it on a disquette
PCLOS

Install now do it soon
Listen to the purr you will love the tune
Never a drag never a snag
That's The PCLOS Rag

Come to the forum hear the fanfare
Meet new friends waiting for you there
Oh how they love to share
PCLOS

Install now do it soon
Listen to the purr you will love the tune
Never a drag never a snag
That's The PCLOS Rag

Short Topix: Anti-Vaxxers Facing Social Media Crackdown

by Paul Arnote (parnote)

NSA Reverse Engineering Tool Ghidra Now Open Sourced

In a move that caught many by surprise, the United States National Security Agency, a.k.a. the NSA, has open sourced their reverse engineering tool, called Ghidra, according to an article on ItsFOSS. Yes, THAT U.S. agency that has vacuumed up everyone's data and denied every living being on the planet a right to privacy.

Written in JAVA, Ghidra runs on Windows, MacOS and Linux. The program will reverse engineer computer programs and output compatible C code to accomplish the same tasks. In essence, it's a decompiler. At the time of the release announcement, it was unknown which CPU architectures it was able to reverse engineer programs for. It is thought to – at least – work for x86, x64 and ARM architectures. While the program has a GitHub page that is currently serving as a placeholder page for the code, you will still (at this time) have to visit the aforementioned link to download the actual program.

There is an interesting discussion of the tool on Reddit. Joining in on the Reddit discussion are several who, in whatever their former capacity used to be, have used Ghidra in the past. The information they provide is quite telling about its capabilities. The commenters rank Ghidra right up there with high level commercial decompilers, like IDA Pro, Hex-Rays Decompiler, Binary Ninja, and Radare2.

So why open source the tool now? Well, that's an area of speculation. The public knows of the tool's existence, thanks to the leak of Vault 7 documents from the CIA on WikiLeaks in March, 2017. Open sourcing the tool will allow the open source community to not only help maintain it (free of charge, nonetheless), but the open source community can also help grow and enhance the tool's capabilities.

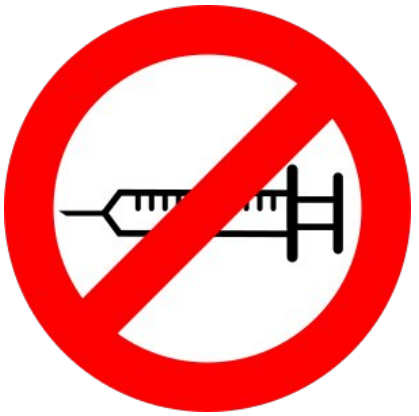
ANYTHING dealing with or coming from the NSA isn't without controversy. Not only is the agency one of the most hated in the world, but it is also one of the least trusted agencies the world over. Some commenters openly worried about back doors or "phone home" routines in the Ghidra software. The distrust and distaste for the NSA is certainly well earned by the agency.

As far as Linux is concerned, this isn't the first time NSA-born software has appeared. Back when Linux kernel 4.17 was being released, there was a fervor that swept among Linux users over the inclusion of the NSA encryption algorithm called Speck. Speck is a lightweight block cypher that Google wanted to include and use to encrypt Android files, via dm-crypt and fscrypt. Its original push was to be used as lightweight encryption of IoT devices.

As is usual, much suspicion surrounded the inclusion of Speck (and its companion cypher Simon) in the Linux kernel. Due to its history of abuse, any move by the NSA is met with intense suspicion and scrutiny. Speck was rejected by the ISO committee, due to concerns that the NSA had placed "backdoors" into the code, allowing spies and government agencies easy access to "encrypted" data. The inclusion of Speck was reportedly removed by Linux kernel 4.19.

In a 2013 article on The Register, it was reported that Linus Torvalds admitted to having been approached by U.S. "government men" to include a "backdoor" in Linux. He made this admission at a New Orleans LinuxCon question and answer session. He did so by shaking his head yes while supplying an emphatic NO as an answer to the question. He then shook his head NO and emphatically said NO, to answer the question of whether a backdoor exists in Linux.

Anti-Vaxxers Facing Social Media Crackdown



There's one topic that is extremely polarizing: vaccines. The vast majority of people support the

routine administration of vaccines. But an ever growing group of people are buying into the “vaccines cause autism” argument. This is despite repeated studies by numerous healthcare organizations around the world that all show there to be no link between vaccines and autism. The World Health Organization (WHO) lists "vaccine hesitancy" as one of its 10 threats to global health in 2019.

But, the latter group, commonly referred to as “anti-vaxxers,” has found a vehicle for their message on social media. After recent outbreaks of measles in various pockets throughout the U.S., the AMA (American Medical Association) and the AAP (American Academy of Pediatrics) have urged the CEOs of Facebook, Google, Pinterest, Twitter and YouTube to take steps to address vaccine misinformation on their sites.

As a result, YouTube has demonetized anti-vax videos. Facebook will demote pages and groups that share anti-vax information. The company will also stop search results for anti-vaxxer pages on its app and website. Similarly, anti-vaxxer content will no longer appear in Instagram's Explore page, or in the app's hashtag pages. Pinterest will no longer return search results for items related to vaccines, whether they are pro-vaccine or anti-vaccine views.

The anti-vax movement gained considerable steam after then British physician, Dr. Andrew Wakefield, published a paper in the British medical journal Lancet that linked vaccines to autism and bowel disease. In the 20+ years since, Wakefield has been discredited, has had the paper’s “co-authors” rescind or withdrew their support for the “study’s” interpretations, and has been “struck off the UK medical register for unethical behaviour, misconduct and dishonesty for authoring a fraudulent research paper that claimed a link between the measles, mumps and rubella (MMR) vaccine and autism and bowel disease.” Additionally, “Wakefield is barred from practising as a physician in the UK, and is not licensed in the US. He lives in the US where he has a following, including prominent celebrity anti-vaccinationist Jenny McCarthy, who wrote the foreword for Wakefield's autobiography, Callous Disregard. She has a son with autism-like symptoms that she is convinced were caused by the MMR vaccine. According to Sunday Times of London’s investigative reporter Brian Deer, as of 2011, he lives near Austin (Texas) with his family.”

While most would argue that these moves by the tech giants are the best moves and in the best interest of the community at large, this does present a “slippery slope” type of situation. Do we squelch the views and voices of those expressing ideas that are different from our own, or that are not in line with the majority opinion du jour? At what point do we draw the line? Indeed, this is a very difficult situation. If public opinion shifts, how long will it be before YOUR views and voices come under attack?

YouTube’s mission statement starts out, “Our mission is to give everyone a voice and show them the world.” While they are not taking down the anti-vaxxer videos, they have demonetized them. YouTube has recently came under fire for similarly squelching the voices of conservative and populist viewpoints, while promoting more progressive and leftist viewpoints.

Facebook has similarly been accused of the exact same thing in the not too distant past, and the complaints continue to this day. Facebook has even enlisted the infamous “Snopes.com” website to “fact-check” information presented on Facebook, despite Snopes being widely perceived as having or supporting decidedly leftist and progressive viewpoints.

Yes, it is a slippery slope, indeed.

A Password Free Web?

WebAuthn

The World Wide Web Consortium (W3C) and the FIDO Alliance today announced the Web Authentication (WebAuthn) specification is now an official web standard. This advancement is a major step forward in making the web more secure— and usable—for users around the world.

W3C's WebAuthn Recommendation, a core component of the FIDO Alliance's FIDO2 set of specifications, is a browser/platform standard for simpler and stronger authentication. It is already supported in Windows 10, Android, and Google Chrome, Mozilla Firefox, Microsoft Edge and Apple Safari (preview) Web browsers. WebAuthn allows users to log into their internet accounts using their preferred device. Web services and apps can — and should—turn on this functionality to give their users the option to log in more easily via biometrics, mobile devices and/or FIDO security keys, and with much higher security over passwords alone.

“Now is the time for web services and businesses to adopt WebAuthn to move beyond vulnerable passwords and help web users improve the security of their online experiences,” said Jeff Jaffe, W3C CEO. “W3C's Recommendation establishes web-wide interoperability guidance, setting consistent expectations for web users and the sites they visit. W3C is working to implement this best practice on its own site.”

A user-friendly solution to password theft, phishing and replay attacks

It's common knowledge that passwords have outlived their efficacy. Not only are stolen, weak or default passwords behind 81 percent of data

breaches, they are a drain of time and resources. According to a recent Yubico study, users spend 10.9 hours per year entering and/or resetting passwords, which costs companies an average of $5.2 million annually. While traditional multi-factor authentication (MFA) solutions like SMS one-time codes add another layer of security, they are still vulnerable to phishing attacks, aren't simple to use and suffer from low opt-in rates.

With FIDO2 and WebAuthn, the global technology community has come together to provide a shared solution to the shared password problem. FIDO2 addresses all of the issues with traditional authentication:

* **Security:** FIDO2 cryptographic login credentials are unique across every website, biometrics or other secrets like passwords never leave the user's device and are never stored on a server. This security model eliminates the risks of phishing, all forms of password theft and replay attacks.

* **Convenience:** Users log in with convenient methods such as fingerprint readers, cameras, FIDO security keys (typically plugged into an available USB port), or their personal mobile device.

* **Privacy:** Because FIDO keys are unique for each Internet site, they cannot be used to track you across sites.

* **Scalability:** websites can enable FIDO2 via simple API call across all supported browsers and platforms on billions of devices consumers use every day.

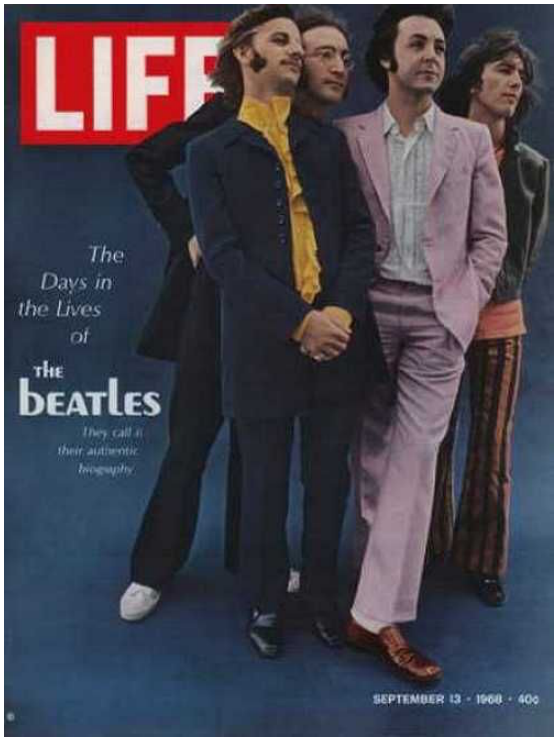
"Web Authentication as an official web standard is the pinnacle of many years of industry collaboration to develop a practical solution for stronger authentication on the web," said Brett McDowell, executive director of the FIDO Alliance. "With this milestone, we're moving into a new era of ubiquitous, hardware-backed FIDO Authentication protection for everyone using the internet."

Getting started

For services providers and vendors ready to get started with FIDO2 specifications and browser/platform support, the FIDO Alliance has provided testing tools and launched a certification program. Currently, there are many FIDO2 Certified solutions available to support a wide variety of use cases, including FIDO Certified Universal Servers that support FIDO2 and all prior UAF and U2F devices for full backward compatibility with the full range of certified FIDO authenticators.

Visit the FIDO Alliance website for more information on FIDO2, including resources for developers and product vendors interested in taking part in the FIDO Certified program.

50 Years Later: Boy Who Stole Life Magazine With Beatles Cover, Returns It To Library



Fifty years ago, at the height of Beatlemania, a young Ohio boy named Brian saw a copy of Life magazine. On its cover was the Fab Four. He stole that copy of Life magazine to make it his own.

But somewhere around the end of February and the beginning of March, he decided to give that stolen copy of the magazine back to the library. He sent it to them, along with a note that read:

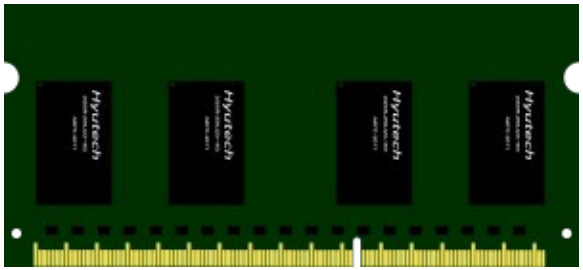
"Hello. I stole this magazine from the Parma Ridge Road library when I was a kid. I'm sorry I took it. I've enclosed a check for the late fee."

Accompanying the note was a check for $100, despite that fact that over five decades have passed. Even though his "real" fine would have been somewhere in the neighborhood of $1,800, the library caps the maximum fine at $100, which he remitted.

The library took the time to write Brian back:

"To the Beatles fan who "borrowed" this copy of *Life* magazine in 1968: Thank you for returning it this week and clearing your conscience."

Now Is The Time To Add More Memory Or A SSD



Marketplace factors have made this the ideal time to finally add that extra memory you want to your computer, or to add that SSD you always dreamed of having. Having just purchased a used laptop from eBay that had a SSD installed in it, I can vouch for the speed increase running PCLinuxOS. The laptop,

with an Intel i5 processor and 8GB RAM literally is screaming fast, especially when coupled with my favorite lightweight desktop, Xfce.

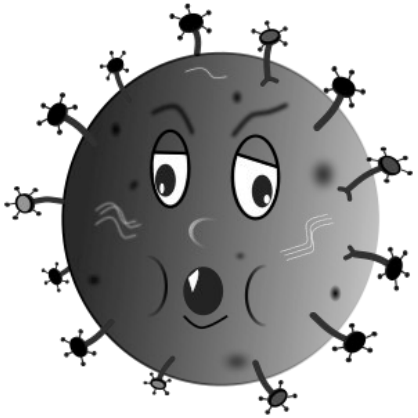
According to one TechRepublic article, prices for PC DRAM, the memory chips that make up (SO)DIMM memory modules that we use in our computers, have fallen 30% in the first quarter of 2019. This represents the biggest decline in prices since 2011. The falling prices are a direct result of Intel not being able to keep up with demand, as they moved from a 14nm process to a 10nm process. The transition has not been particularly smooth for Intel, prompting the delays in shipping adequate numbers of Ice Lake CPUs to keep up with demand among computer manufacturers. As it stands now, DRAM suppliers are sitting on six weeks worth of inventory.

At the consumer level, buyers can expect to see significant discounts on current, top-line memory products. For example, a 32GB DDR4 SODIMM module currently runs around $225, down from the $300-$450 prices seen a year ago. DDR3 memory modules aren't likely to see as much discount, since most manufacturers are focused on producing the newer memory configurations.

Meanwhile, NAND flash memory, used in the construction of SSDs and flash memory cards, has seen its prices fall pretty dramatically, according to another TechRepublic article. Probably the most dramatic example of this is that you can now buy a 512GB SATA SSD for the same price that you could have purchased a 256GB SATA SSD a year ago. Because of the lower prices, computer market analysts expect SSD and NVMe devices to fully occupy half of the market share by the end of 2019.

If/when your replace your rusted platter HD with a SSD, prepare to say hello to longer battery life (thanks to lower power consumption of SSDs when compared to traditional HDs), and faster overall computer response (thanks to memory reads being faster than reading data from a rotating platter of rust). The full potential of SSD and NVMe devices won't be realized, however, until the throughput rate of the SATA interface is increased. Current SATA throughput rate is currently limited to approximately 550MB/s, far slower than either device is capable of producing.

Want To Prevent Catching The Flu?



At least here in the U.S., this past flu season has been a lot milder than last year's. The annual flu vaccine given at the beginning (or slightly prior to) the current flu season was a quadrivalent vaccine that included protection against Influenza A H1N1, the predominant strain that was responsible for influenza in the early part of the flu season. This is the same strain of influenza that was responsible for the 1918 Spanish Flu epidemic. Over the years, H1N1 appears to have mutated to become less severe and less virulent.

However, the late flu season has seen a rise of H3N2 Influenza A cases – and which was not included in the quadrivalent annual flu vaccine. Be careful to not confuse actual influenza with what most people call the "stomach flu." The former is a respiratory virus, and causes severe viral pneumonia in its victims. The latter is a gastrointestinal virus, complete with fever and muscle aches, but causing intense vomiting and diarrhea. The latter is typically caused by norovirus, which is usually the same culprit for all those people getting sick on cruise ships that you hear about. It typically lasts 24 to 72 hours before running its course.

So, how do you help protect yourself from contracting the flu? Keep in mind that the virus can be breathed in when someone in close proximity to you coughs or sneezes (as it's very easily spread via airborne droplets). BUT, it can also live on surfaces, and if you come into contact with those surfaces and then touch your nose, eye(s) or mouth, you can still become infected. Hugging, kissing and shaking hands can also facilitate the spread of influenza from person to person.

That is why the Washington Post ran an article detailing that merely washing your hands frequently for ***at least*** 20 seconds can go a long way to helping prevent catching influenza. This means PROPER hand washing. The five-second get them wet and pray method will NOT work. So, let's review proper hand washing.

First, 20 seconds is about how long it takes for you to sing the ABC's song at a casual pace. Or, just sing the "Happy Birthday To You" song – twice. Second, use plenty of soap and insure that you are soaping and rubbing the fronts and backs of your hand, around your nails and cuticles, and between your fingers. It doesn't matter if you use hot or cold water. If water isn't available, alcohol-based hand sanitizer will work just fine.

Roundup At The PCLinuxOS Corral

LAST MONTH, we told you about some security issues with **Amazon's Ring doorbell**. Well, one month later, the Jerusalem Post has exposed further security issues with the device. In the latest news, video of

someone you know or expected at your door can easily be inserted, replacing the video of the actual person standing at your door. The fault is the same as we reported last month: Ring's unencrypted video feed. If that doesn't scare the bejeezus out of you and make you leary of using these unsecure devices, then nothing will.

Some **hackers** actually make their living chasing down bugs in programs. There are hackers that have made well over $1 million per year by cashing in on software bug bounty programs. So, one TechRepublic article asked the question: do bug bounties help open source security? The answer, was yes.

Hey … what about that old **USB flash drive** you have? Have you ever purchased a used flash drive? Have you ever lost a flash drive you typically carry in your pocket? You better be careful, because the contents of that flash drive can reveal a lot more about you than you think. In fact, 67% of used flash drives contain recoverable data from its previous user, according to a study commissioned by Comparitech, and conducted by University of Hertfordshire researchers. They purchased 200 used flash drives from eBay, second hand shops, and live auctions – 100 from the UK and 100 from the U.S. So what did they find? They found intimate, private and sensitive files, including nude photos, business documents, wage slips, business documents, private memos, tax statements, receipts, and medical records, among other things. The researchers only used publicly available software that can be downloaded from the web.

So, what can you do to secure your data? First and foremost, encrypt the data on your flash drive. The inconvenience may be worth it. This way, should you ever lose it, you won't have to worry about your data falling into evil hands. If you're looking to sell a previously used flash drive, take the time to irreversibly erase all previously stored data. Fortunately for Linux users, this is a simple task of running the dd command on the drive. Use the following format for the command: **dd if=/dev/urandom of=/dev/sdX bs=1M**, where sdX is the drive designation of the flash drive. When done properly, there is less than 0% chance of retrieving data that was stored on the drive.

Screenshot Showcase

Posted by Old-Polack, March 21, 2019, running KDE.

Inkscape Tutorial: Stylized Text

by Meemaw

I came across this tutorial a few weeks ago. This is another method for decorative text, and pretty fun. I also chose it because it includes keyboard shortcuts for customizing your text. In some instances, rotating and moving individual letters will give your text a different look that works better for your project than straight text would.

Open a new page, then choose your font and do your typing. Fat or heavy text will work better for this - I used a font called Flubber.

Now we need to move &/or rotate the individual characters. You can easily use the keyboard shortcuts as described below. Note that your cursor will have to be placed to the left of whichever letter you want to change, and if there are more letters in that line, your change may affect them all.

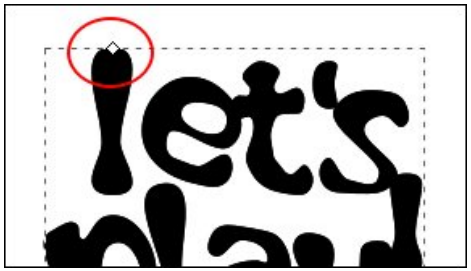
Manual tracking is the distance between each letter. Using **Alt-Right** or **Alt-Left**, you can apply a shift of one unit left and right. **Shift-Alt-Left** and **Shift-Alt-Right** will apply a shift of ten units each direction.

Rotation can be performed on each letter individually. **Alt-[** and **Alt-]** can be used for counter and clockwise rotation around the lower left character anchor point.

Adjusting baselines will move the lines of text closer together. **Alt-Up** and **Alt-Down** are the keyboard shortcuts. Note that your anchor will be the first character in the text to the right of the cursor, so adjusting the first character will shift the entire line up or down accordingly.

I chose to space the letters out just a bit, rotate a few of the letters, and adjust the baseline on most of them. Playing with my text produced this:

Next, select your text, then use the **Linked Offset** tool from the **Path** menu. This will create the single node handle to drag outward. Do not change your original text to a path.



Grab the handle and pull it out, making the text look heavier. Change the colors on the two different text items, so you can see what you're doing. As you can see below, the white text is the original, and the blue is the offset text.

Select the original text object and give it a stroke of another color that works with your palette. It's your project, so pick the colors you like, and colors that complement each other. Do the same with the outset text.

Choose the outset object and use **Filters > Shadows and Glows > Drop Shadow** to add your drop shadow. Choose your original text and do the same. You can see two samples below.

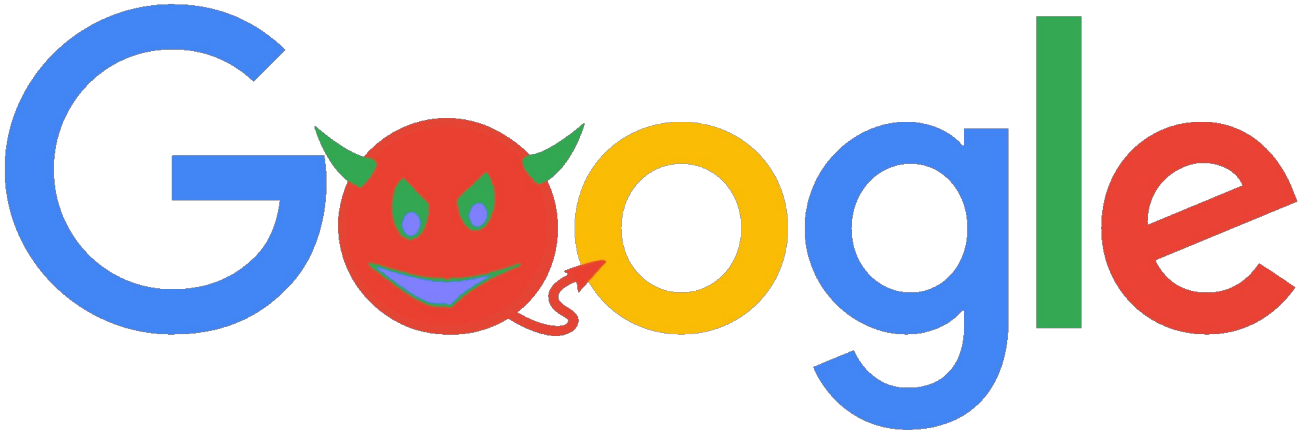
Eliminating Google From Your Digital Life

by Paul Arnote (parnote)

When Google was first started in 1998, its motto was "do no evil." Now, 20+ years later, it has become the embodiment of evil for many, many people. For better or worse, they did it to themselves. It is, in many ways, the worst of self inflicted wounds.

The larger Google grows, so grows the list of grievances against it. Probably chief among those complaints is how it is "in bed" with the NSA, GCHQ, and all the other three and four letter spy agencies around the world. Right behind that is its disregard for your privacy. Google knows ***everything*** about not only your digital life, but pretty much your entire life. It knows where you've been, who you've phoned, what you've purchased, what you've viewed online, where you eat, where you sleep, what you like to read, what you like to watch, where you work, when you work, the routes you travel to and from work, etc., etc. There's little that Google doesn't know about you. Yes, their information about YOU is THAT pervasive and complete.

Never mind the fact that despite using Linux as their default operating system, and despite using Linux as the core of their mobile platform (Android), they have given very little back to the Linux community. When Google SketchUp was around, Linux users only got a Wine-packaged Windows version of the software to run. When Google Picasa was around, Linux users were met with the exact same insult with the Picasa desktop app. Despite promises long, long ago, Google has NEVER released a Linux version of their Drive desktop client software, and even after it was revealed that they have a Linux version of their Drive desktop client that is used internally on a very regular basis.



Google is everywhere, and has its tentacles in virtually everything we do online. Obviously, it IS the search engine of choice for many, many users. The name Google has even become a verb, as in "just Google it." It runs/owns Gmail. It owns YouTube. It has its fingers in providing blogging sites, cloud storage, photo storage, news aggregation, weather information, shopping, gaming, phone service, internet service, television service, mapping services, merchant rating/ranking services, a profitable app store, a free online office suite, a calendar service, books, finance, and many, many other aspects of online life.

Google's primary source of revenue is from selling advertising. It uses all of that information it gathers about you and your life to target advertising to the things you like or have expressed interest in. Reportedly, Google never sells the information it collects about you to third parties. Google is quite good at collecting your data, too, since it's one of the largest and most successful corporations around. Their pursuit of profits supercede your rights to privacy, or any other rights you might have, at least in their corporate mindset. Them having their fingers in so many pies allows them to collect more and more and more data about you, and the more data they have, the more specifically they can target ads you might be interested in.

As Google departs from its original motto, there is a growing body of users wishing to depart from Google's increasingly evil intrusion into our daily lives. Some old timers around here might remember an old PCLinuxOS forum administrator, named Hootiegibbon. Hootie has worked to "de-google-fy" himself as much as is possible. There are only a couple of Google services that he uses today that he is unable to (for his own personal reasons) distance

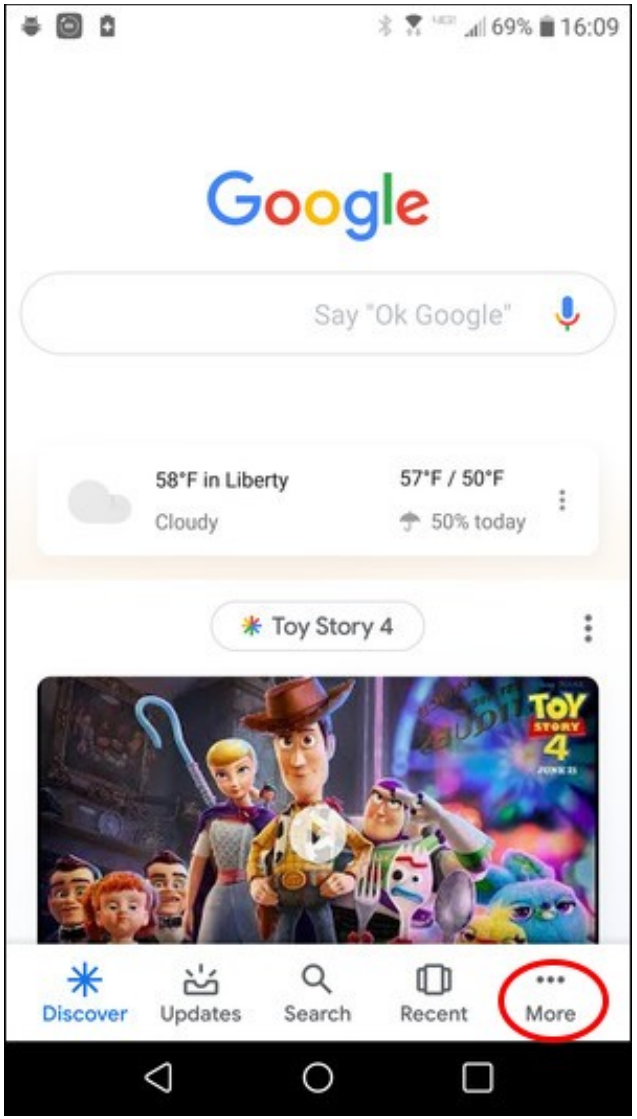
himself from. As for the rest of his Google history, he has, should we say, “disengaged.” He has even found an alternative email provider, giving Gmail the heave-ho about a year ago.

He isn’t alone. I can name several current and former forum members who shun the use of Google services. But, because of Google’s immense diversification, it will be rather difficult for most people to completely eliminate Google from their digital lives. Google is just like the serpent Hydra in Greek mythology, who grew back two heads every time you cut one head off. Each head not only spits poison, but it also helps to feed the monster. And, if you use an Android smartphone or tablet, the odds of getting Google out of (or minimizing its footprints in) your digital life are extremely longer. Android is Google’s mobile platform, and Google’s tentacles are inextricably woven throughout just about everything you do on an Android device.

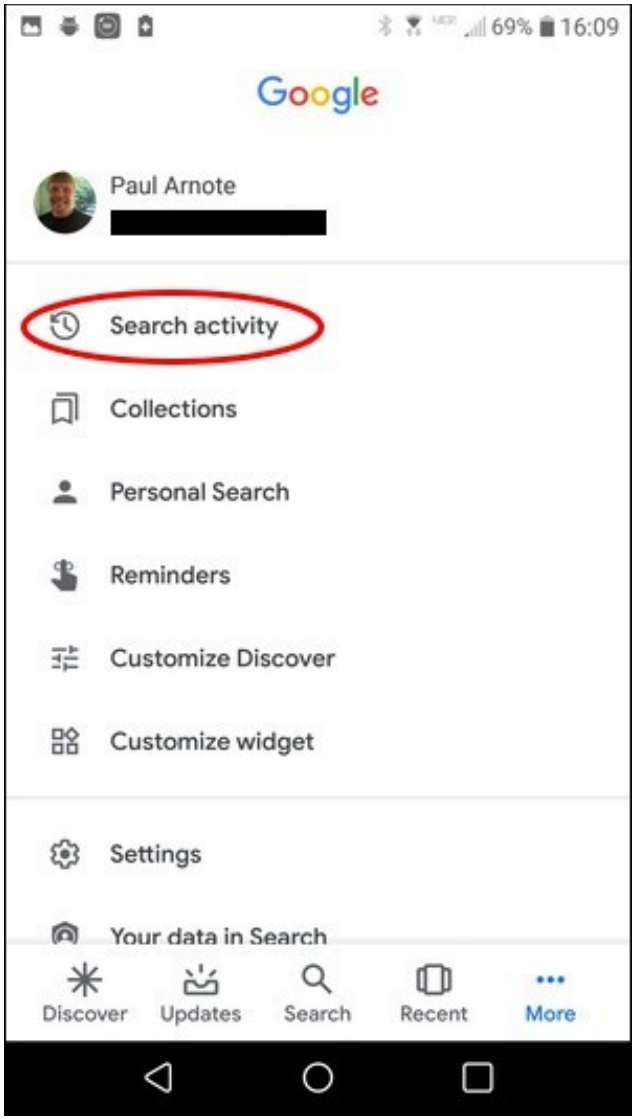
So, while this article is entitled “Eliminating Google From Your Digital Life,” you might not be able to completely extricate yourself from Google’s grasp. At best, you will likely be able to minimize their intrusion into your digital life. Eliminating Google from your digital life is going to be a very, very difficult task, and may be impossible.

The first place to start is by deleting your Google search history. In fact, an old article on CNBC’s website got me thinking about this. Their article centered around using the Google app on a mobile device. I’ll expand here to also include how to do the same thing on the desktop. Whichever method you use, the results will be the same. On the mobile device, your desktop data will also be displayed for a given account. On the desktop version, your mobile data will also be displayed for a given account. So, you can achieve the same results on either platform.

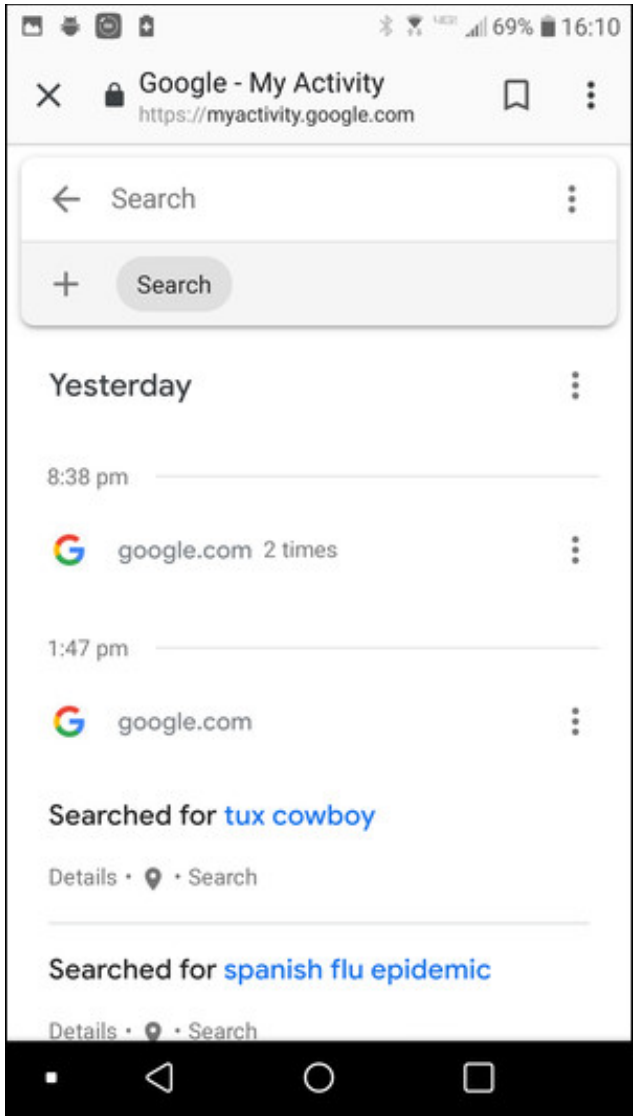
Let’s take a look at the mobile platform first. On your mobile platform, launch the Google app.



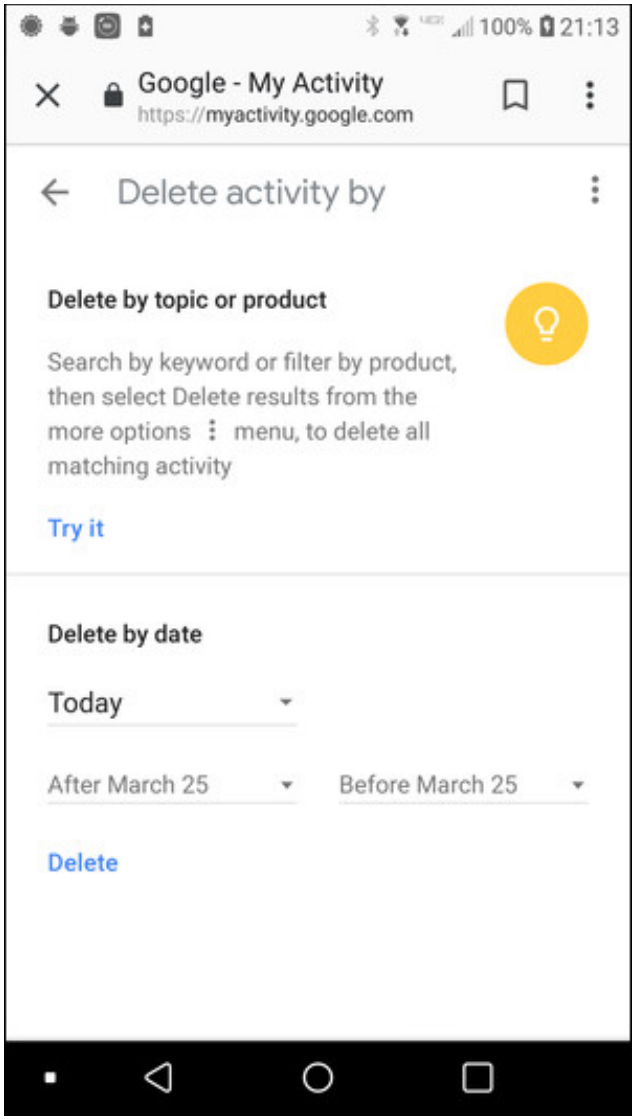
Once you open your Google app, click on the “More” button.



Next, click on the “Search Activity” menu item.



Then, choose the three dot menu that's to the right side of the line that says "Search." Select "Delete activity by..." from the menu.

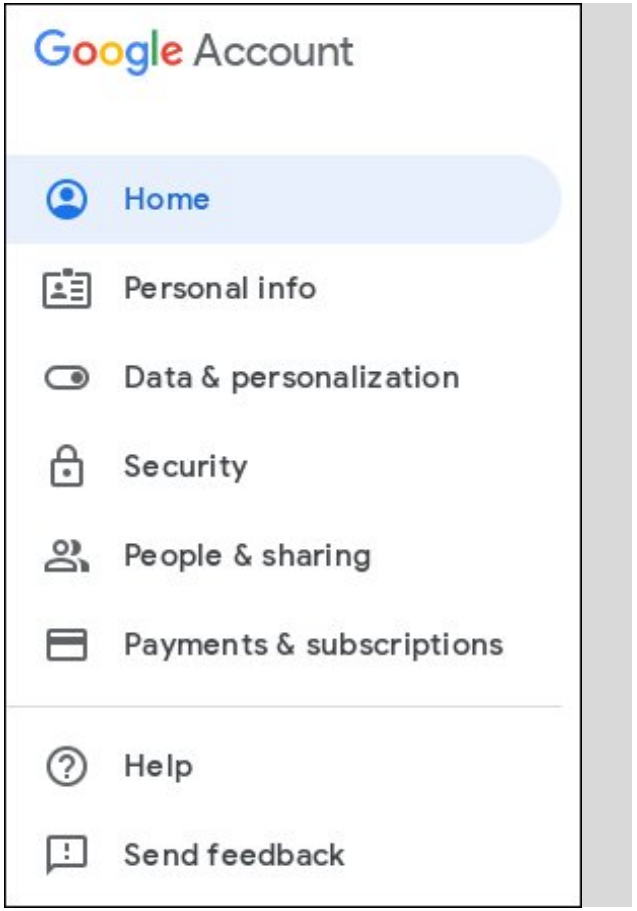


In the lower half of the screen, you will see the "Delete by date" dropdown list box. If you open it, one of the choices is "All time." Selecting this will delete ALL of your search history.

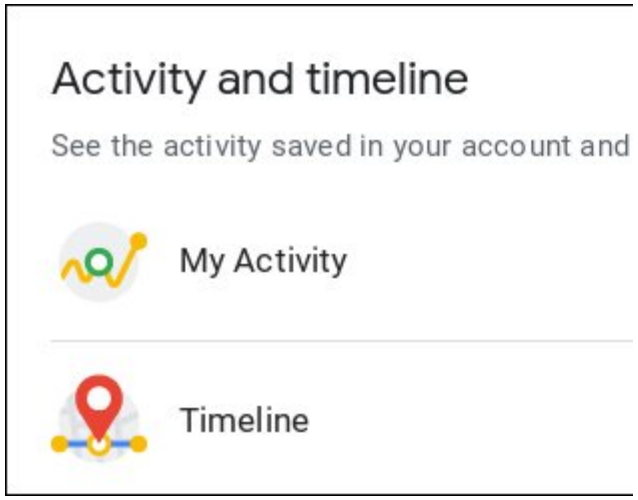
From the same three dot menu that got you to the above screen, you can also choose the "Other Google activity" menu item. This will give you an option to delete all sorts of items on individual Google services. There's a lot to go through on that screen, with many, many options available.

Now, when you do the same thing on the desktop, you have more options. In the end, though, the same objectives are achieved.

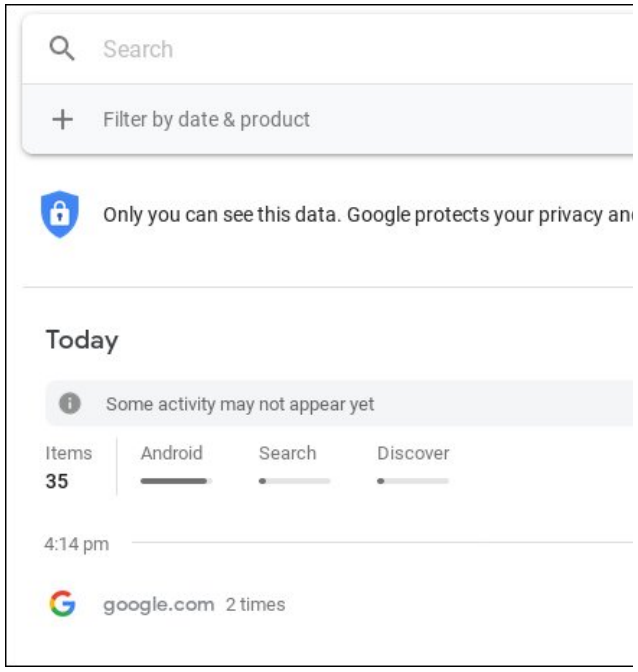
To get started, click on your login logo in the upper right corner of the Google homepage. Select the "Google Account" button in that popup window.



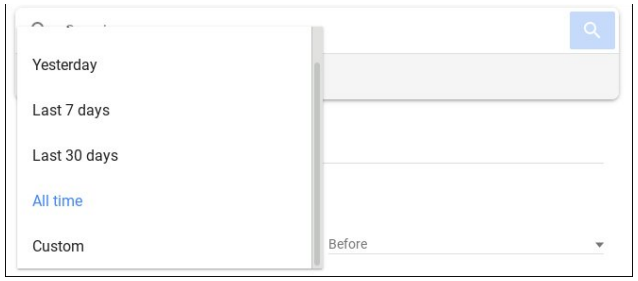
From the menu that appears on the left side of your screen, select the "Data & personalization" option.



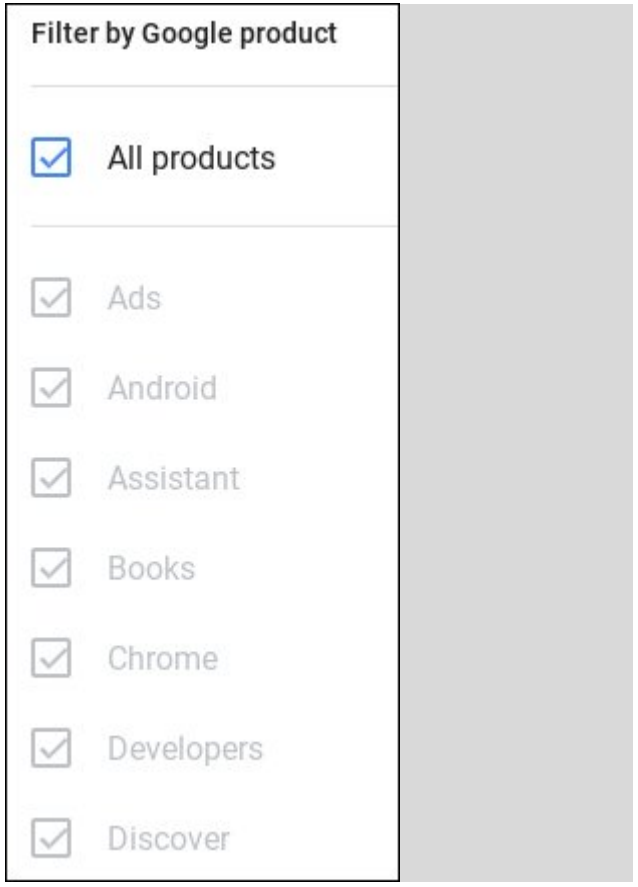
The third "section" down among the choices is the "Activity and timeline" section. Click on "My Activity."



Click on the "+" sign next to "Filter by date and product."



Under "Filter by date," choose what time frame you want to include when deleting your Google history.



Under "Filter by Google product," you should find "All products" already checked. If you choose, you can clear this checkmark, and select individual Google products that you want to delete your history for.

The Search For Replacements

Replacing Google services isn't necessarily going to be easy, given Google's reach and the breadth of their services. But, replacements do exist. You are not alone in searching for alternatives, either. In fact, I wasn't aware of many of these alternatives until I did a search for them for this article.

Search: Believe it or not, there are other search engines out there besides Google. Of course, Yahoo! is still out there and perking along. Yahoo! was recently purchased by Oath, who is owned by Verizon Wireless. Verizon has a very lackluster reputation for protecting the privacy of its users. Then there is Bing, from Microsoft, who also fails to excel in the protection of user privacy (to put it mildly). But there are two in particular that do an exemplary job of preserving your privacy, and provide search results without tracking you. DuckDuckGo and StartPage both fit into this category. DuckDuckGo pulls in search results from several different search engines, while stripping out your personally identifiable information. StartPage uses Google for its search results, but also strips out your personally identifiable information. It also offers a proxy service, so you can browse websites safely and anonymously. If none of these are what you are looking for, FossBytes lists 12 alternatives to using the Google search engine.

Video: There is no denying that YouTube is the largest of the video sites. 300 hours of video are uploaded to YouTube every minute. Nearly 5,000,000,000 (yes, billion) videos are viewed on YouTube every day. In an average month, 80% of adults 18-49 watch a video on YouTube. But, as YouTube has evolved, it has decreased the monetization of videos for those who provide content. It has also been accused of having a heavy hand when it comes to censorship. But, alternatives do exist. DailyMotion is one such service. It has a similar layout to YouTube, making navigation easy. It does limit the uploading of HD videos to Pro level users, and there is a 4GB file size limit on videos

(about 60 minutes). If you can work around these limitations, then it provides a viable alternative to YouTube. Vimeo is another popular alternative that strives to feature videos without all the distractions. It's not perfect, though, limiting uploaders to a maximum of 500MB uploads per week. You can "upgrade" to a maximum of 5GB uploads per week, for a monthly fee. Metacafe focuses on short videos, with a limit of 90 seconds. Vevo is the place to go for music videos. The Internet Archive offers up all sorts of old, off-beat videos that you probably can't find anywhere else. Included there are lots of Three Stooges videos, Charlie Chaplin movies, and a LOT of "B" SciFi films. There are others, and TwitGoo compiled a list of 15 YouTube alternatives, if none of these fit what you're looking for.

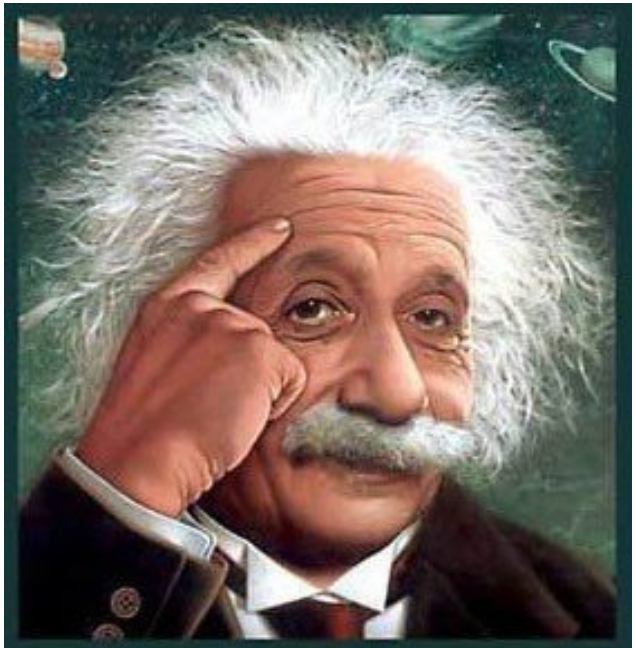
Office Suite: This one is a little more difficult to find a replacement for. Of course, you can always use LibreOffice and share files back and forth. But, if you're looking for a collaborative environment, LibreOffice isn't necessarily going to be the easiest avenue to travel. You could use Office 365, but there are two drawbacks to it. First, it's Microsoft. Second, you will have to pay ($70 U.S. for an annual subscription, personal use only). For a free alternative (Google Docs is also free), look to ONLYOFFICE, which is a nice Google Docs alternative. You will have to set up an account, but it does a nice job as a Google Docs alternative. Zoho Docs is another free alternative that behaves almost exactly as Google Docs. In fact, when the magazine staff was experiencing issues with a change in Google Docs a few years back, we briefly used Zoho Docs until the Google Docs issues were straightened out. There are a few others that may or may not fulfill your needs, but Beebom reviewed 10 of them. Most are free, but some require payment.

Summary

While not all-inclusive, this list of alternatives should get you well started on your way to at least lessening your reliance on Google for common, daily online activities. Of course, there are other ways to lessen Google's grip on your digital life. For example, PCLinuxOS users can get a free email account at PCLOS-Mail. If you maintain a blog, you can find web hosting for your personal web page, and run your blog with WordPress or one of the other web content management software packages. The price would, for many, be worth it for the peace of mind. For photo storage, you could use Flickr instead of Google Photos, or the image sharing site hosted by YouCanToo that's only for PCLinuxOS users.

Then, of course, there's the alternative of doing nothing at all, and resigning yourself to using Google's services. Sometimes, that is the best choice, dependent on your needs and your situation. As always, the choice is yours.

Meanwhile, don't feel bad for Google that users are trying to escape their grasp. There are always those who don't care one iota about what Google does with their data, and those users will always exist. There will always be new, naive users who are ignorant about such topics. But whatever the case, Google created this situation themselves with their cavalier and uncaring attitudes towards user privacy. If you are privacy-aware, then the decision rests only with you as to whether you can tolerate Google's attitudes towards your privacy.



It's easier than E=mc²
It's elemental
It's light years ahead
It's a wise choice
It's Radically Simple
It's ...

PCLinuxOS
Radically Simple

Screenshot Showcase

Posted by Mr Cranky Pants-YouCanTo, March 2, 2019, running KDE.

OpenVPN File Manager: Manage Your OpenVPN Connection Files

by Bill Dawson (BillDjr)

While I originally created this app for my own use, I was advised that others may find it useful as well. In my travels in Google-town, I noticed there are a vast number of people who need help setting up and configuring their OpenVPN connections. So I expanded and refined this app (referred to as OFM in this article) to help with that, making adding/editing/managing the files involved in using OpenVPN easier, without a lot of file/folder browsing. You still need to use OpenVPN's built-in connection manager to set up your own VPN provider's connection(s). Sorry, but helping with that part is beyond the scope of OFM. However, you will only ever need to do that job once, unless you change VPN providers. OFM provides full backup/restore functionality so that (if the need should arise), you can just restore the required OpenVPN and OFM files and you're set to go … no need to re-configure.

Conventions used in this article and OFM itself

CONF files:

These are the files created automatically when you set up a new connection using the OpenVPN Manager by right-clicking on your panel network icon and selecting "VPN Connection/Manage Connections". They will have a ".conf" file extension, and are created somewhere inside the /etc folder. Your desired CONF connection file will need to be edited for auto-login. If you have trouble locating the appropriate folder the first time, try running a search of your system for "openvpn"...that's how I found mine. Backup/restore is available for CONF files, along with their associated auto-login auth files (ie: auth.txt").

OVPN Files:

These are the files you download that tell OpenVPN how to connect to various servers. They have an ".ovpn" file extension. For the purposes of this app, you must have these files in a folder inside your user folder (ie: /home/bill/openvpn). Your desired OVPN connection file will need to be edited for auto-login. Backup/restore is available for OVPN files, along with their associated auto-login auth files (ie: auth.txt).

PATH Files:

OFM saves the folder locations you input, (See screenshot at the top of this article, "Folders" section), as well as a few others and a "notes.txt" file for, well, any notes you may feel you require. The PATH files are saved with a ".conf" file extension, but are really just one-line text files. You have no reason to, but if you wish to edit these files manually, you can. It's for that reason that I didn't opt for storing your user-data in a database. Backup/restore is available for PATH files and notes.txt

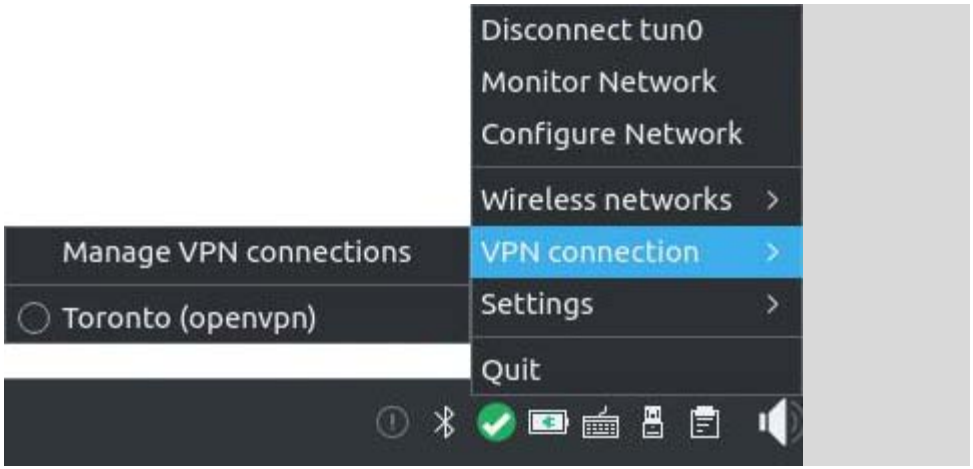
rc.local:

This is a script your system uses during startup. It's the last thing your system will run when it boots up, and is located in /etc/rc.d It is permissible to add your own commands to this file for things you want to automatically run. "rc.local" is not to be edited foolishly and is not to be treated lightly. OFM can/will safely and correctly edit "rc.local" to include an OpenVPN startup command which points to your desired VPN provider's connection file as well as your corresponding authorization file (ie: auth.txt), causing that connection to be established when you boot/reboot your computer. This can be changed and removed completely if you so desire. Please note that since "rc.local" contains the last set of commands to be run at startup, your connection will take a few seconds to activate. Using this function has 2 interesting side-effects:

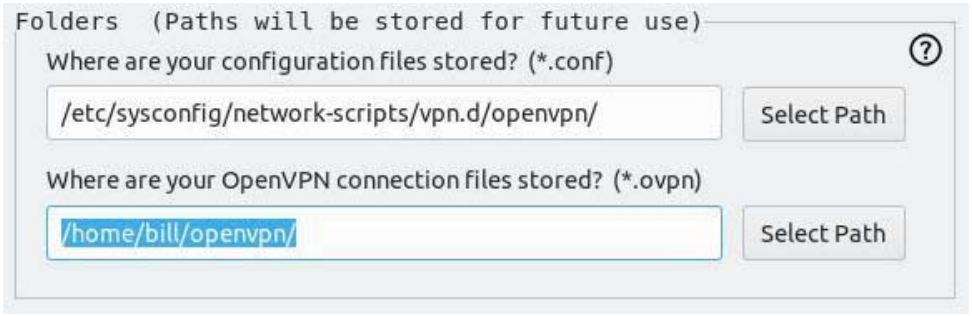
1. You will no longer be prompted for authentication from either the system or your VPN provider for login...OpenVPN will just go ahead and connect.

2. When you right-click your panel network icon for selecting a connection, your auto-connected one will not be marked as active (no colored mark beside its name). Not to worry, your connection is active, and your panel network icon should have changed to indicate this. If you wish to verify the connection, try the link provided below. It should show your location as being that of the VPN provider's server you are connected to.

It should be noted that if you choose not to set up auto-connect, even with auto-login properly set up, you may still be prompted for your VPN username/password when you actually connect. Just click "Cancel". It's an odd quirk.



The first step in using OFM is a critical one. In the top-left corner of the main window, there is a section titled “Folders”, with two text fields and corresponding buttons. These are asking you for two specific folder paths on your system, and once filled, will be saved for future use. You will never have to browse for these paths again. The result of setting these paths is that in every other function, your default file manager (i.e., Dolphin) will open already located at the place you need to be, meaning you can just go ahead and select/work with the file of your choice. DO NOT enter/edit these fields manually.



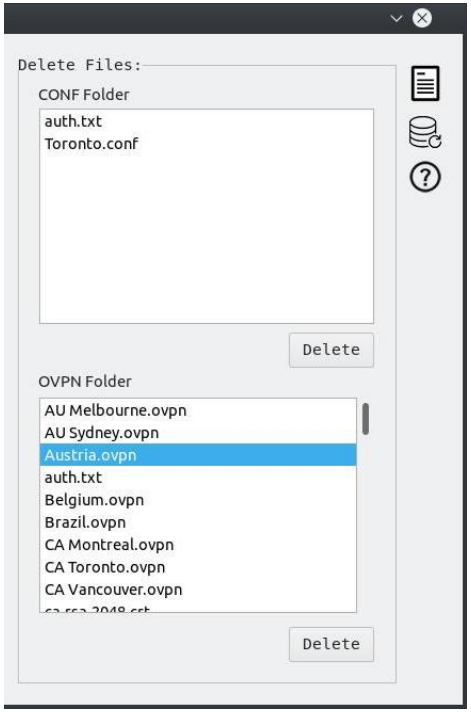
Help windows are available in all sections where they might be required. To access this help, just click the question mark icon in the appropriate section. I have tried to be as informative and as clear as possible.

OFM does not currently have an installer. It is delivered as a simple zip file, and when extracted, will create its own folder named “OpenVPN File Manager”. **Make sure it’s extracted inside your user folder** (ie: /home/bill). Shortcut icons can be found inside the folder.

Using OFM, you’ll detect a noticeable lack of confirmation dialogs. You won’t be told your file was saved. The editing window will just close. Why did I do this? I prefer to just get on with using an app, not click a bunch of message boxes for everything I do. Trust me, the functions work (if they fail, you’ll see a system error message). I’ve tested every little detail to death while creating each one, and well after the project was complete. For better or worse, that’s how it is. If people ever actually use OFM and want those dialogs in place, I’ll add them if a reasonable amount of demand is present.

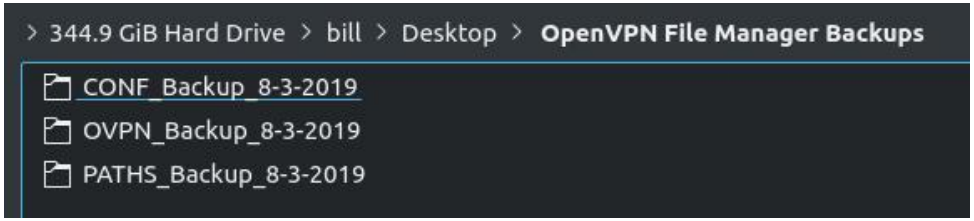
OFM requires superuser access for some of its functions, and needs to use terminal commands for those functions (although you will never see a terminal window). Since the IDE (Xojo) I used to create this app does not allow such access, I was forced find a workaround. This means you will need a tiny package from the repo called “gksu”. All this package does is allow commands to run as superuser. You will, of course, be prompted to grant this access on OFM startup. If you are not comfortable with installing gksu, then OFM will lose half of its functions, and will also ask you to install it as it won’t be detected on your system.

Most of the functions are aimed at creating/editing the files needed for connection auto-login and auto-connecting at boot-time, but if for some reason you wish to make edits to default OpenVPN files for other reasons, that is possible as well. Do so at your own peril.

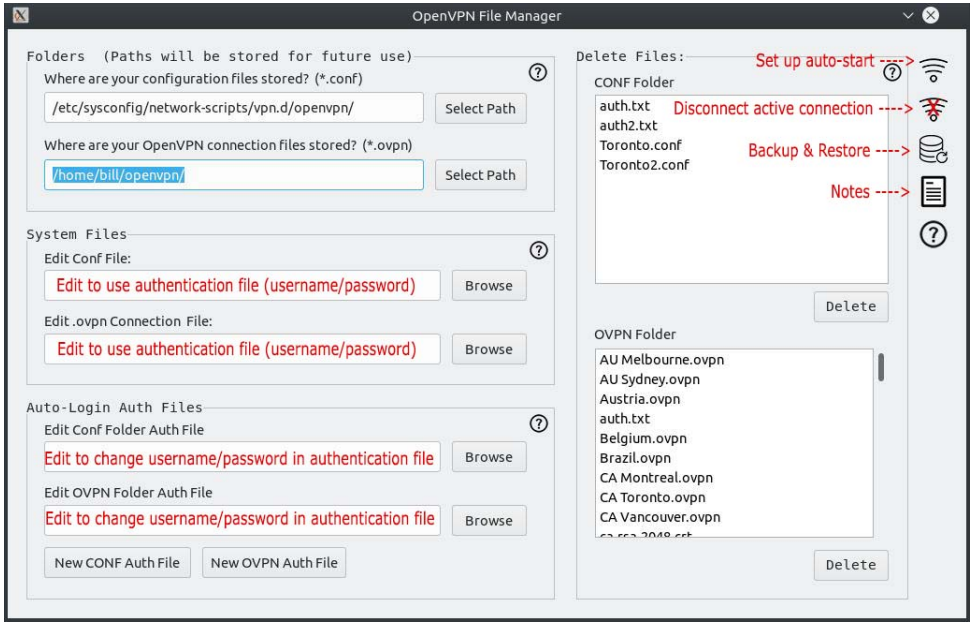


There are two displays that show you the files located in the folders OpenVPN requires (as defined by you in the “Folders” section). From these displays, you may delete files you don’t want or need. Just select the desired file and click “Delete” (see screenshot on the left).

I have also included the ability to backup all related files to a folder on your desktop named “OpenVPN File Manager Backups”. Each file type (there are 4) will be saved into its own folder within “OpenVPN File Manager Backups”, and will include the current date in the folder name. You can also restore these files. When restoring, it would be beneficial to place your backup folder on your desktop, since OFM will open your file manager (ie: Dolphin) at your desktop location for easy selection.



"And now for something completely different." (any Monty Python fans out there?)



This is the main window you will see when you open the app. As stated above, the first thing you need to do is look at the Folders section (top-left). Click "Select Path" for both fields and select the appropriate folders. Configuration files are CONF files, OpenVPN connection files are OVPN files. These paths will be saved as described above, and you will not need to re-select these paths in the future.

The "System Files" section deals with editing the two respective OpenVPN file types. As stated above, this is primarily for editing the line pertaining to auto-login, but other edits you deem necessary can also be made. To edit for auto-login, open the desired .ovpn and/or .conf file.

Find the line (14) that says :

```
auth-user-pass
```

Change it to:

```
auth-user-pass your_authfile.txt
```

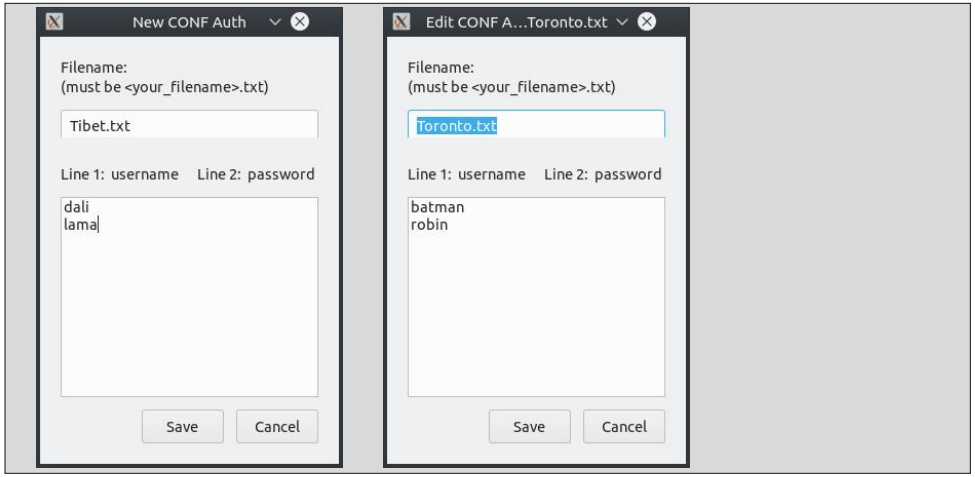
Where "your_authfile" is the actual name of your auth text file.

Do not include quotes, and take note of the single space between "auth-user-pass" and your file name.

NOTE: You should name your auth file so it clearly belongs with your CONF file. For example, if your connection CONF file is "Toronto.conf", then name your auth file something like "Toronto.txt" or "Toronto_auth.txt".

This takes us to the "Auto-Login" section. Here you can create/edit your auth files required for auto-login. These files are related to the above text pertaining to the "System Files" section (ie: your_authfile.txt). When you click the "Browse" button to edit either the CONF or OVPN auth files, your file manager will open in the right place. Just select the desired file and click "Open". A new window will open with an editor in it. Your selected file name will be shown as well as the current text content of the file. Edit as needed and save.

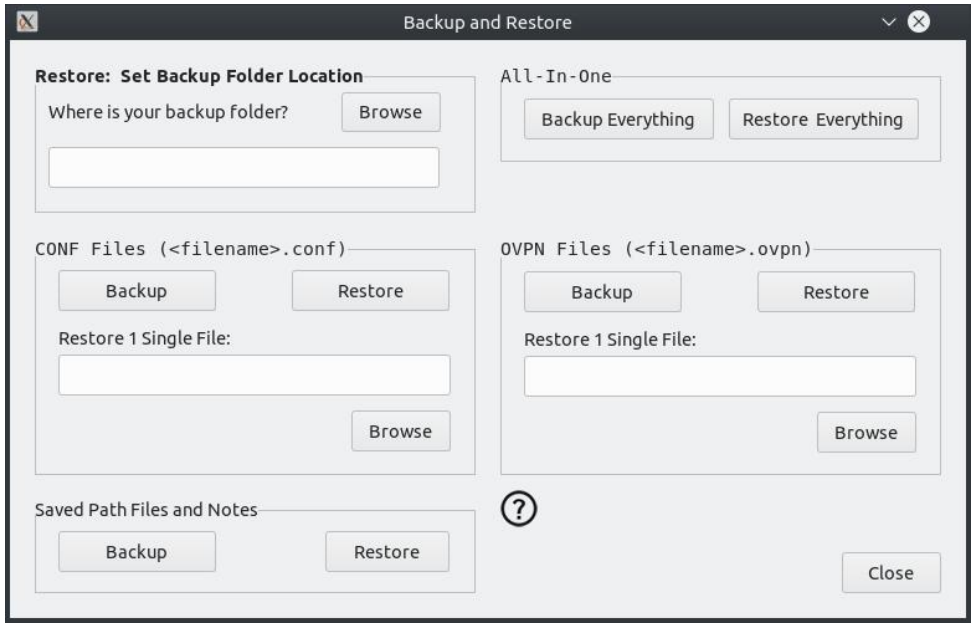
The "New CONF Auth" and "New OVPN Auth" buttons will open the same editor window, but this time you will need to enter your desired file name (MUST be a .txt file extension), plus your username (line 1) and password (line 2) in the text area below the file name. If you are **editing** an auth file, don't bother trying to change the file name, as that change will not be saved. Instead, make a new auth file and then delete the one you don't want (Delete Files section). See screenshots below:



If you create a **new** auth file, the new file name will appear in the "Delete Files" section's relevant file listing.

NOTE: The items in the "Delete Files" section's file listings also include all of your auth files, and can they can be deleted just the same as CONF or OVPN files.

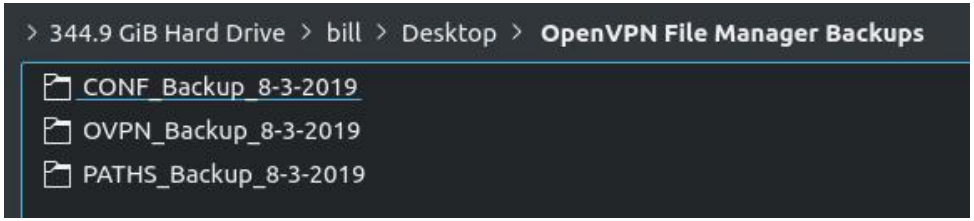
Backup & Restore



As you can see in the screenshot above, there are five sections, three of which are devoted to the backup & restore of the two OpenVPN file types (CONF & OVPN) as well as the saved path & notes.txt files created by OFM ("Saved Path Files and Notes" section).

I chose to backup all files in each section to make things as simple as possible. No matter which section you use (or all of them), a single folder will be created on your desktop named "OpenVPN File Manager Backups". Inside this folder, each section's files will be saved into that section's own sub-folder, named appropriately with the current date included in the folder name (see screenshot below).

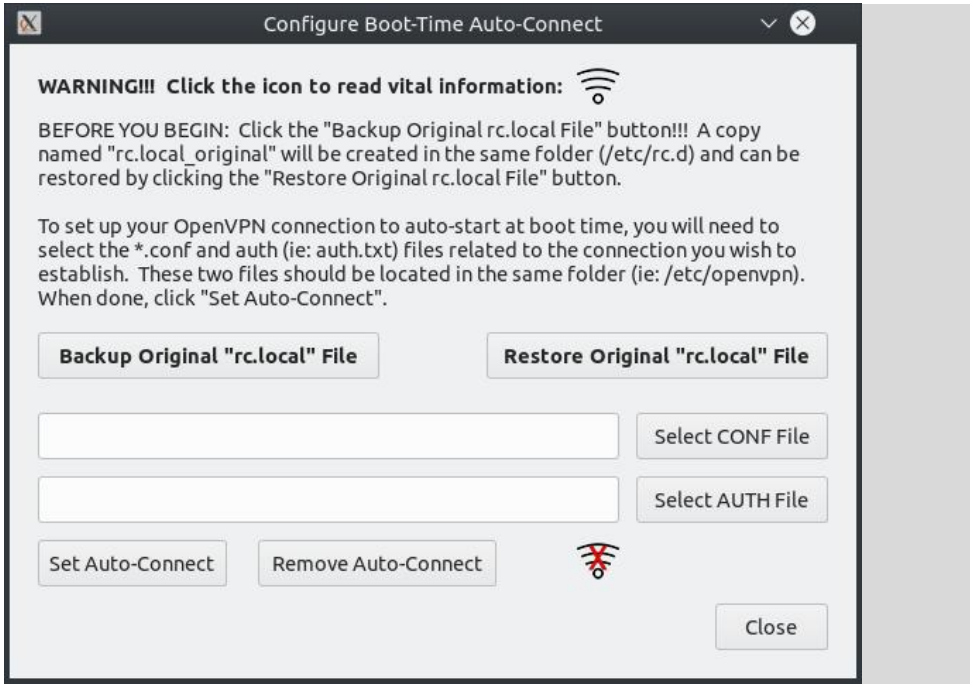
To make things easy for yourself when restoring files, place your backup folder ("OpenVPN File Manager Backups") on your desktop, as that is where OFM will open your file manager.



NOTE: With the exception of restoring one single file, all other "Restore" buttons, including the "Restore Everything" button, require that you first select your "OpenVPN File Manager Backups" location. This is done is the first section (titled in bold), named "Restore: Set Backup Folder Location". Your file manager will open at your desktop.

You will also notice in the "Backup and Restore" screenshot above that you can choose to restore just one single file in either the CONF or OVPN sections. By clicking "Browse" under "Restore 1 single file", you will select the file to be restored and have it restored all in one operation. Be sure to go into the appropriate backup sub-folder to find the right file (see screenshot above).

Auto-Connect at boot-time:



The screenshot above shows the window used for setting up/editing/removing the OpenVPN command line added to "/etc/rc.d/rc.local" to enable your desired VPN connection during boot/reboot. Clicking the "Select CONF File" or "Select AUTH File" buttons will open your file manager in the correct folder, based on the paths you selected in the "Folders" section of the main window (Remember step 1 at the start of this article?).

When you click "Set Auto-Connect", OFM will look to see if you already have an auto-connect command in your /etc/rc.d/rc.local file. If not, one will be added based on your selected CONF and AUTH files. If there is already a command present, then it will be edited to reflect your choices.

Conclusion

This project has been the biggest challenge I've ever had in my relatively short programming experience, but it was worth all the effort. Appearances are very deceiving. OFM looks so simple and basic, yet the code that drives it is unbelievably complex … at least to me. I learned a great deal with this project, not the least of which is that I really **NEED** to learn python. The IDE I used to create this app has so many shortcomings that don't exist in the Windows version. Lesson learned!

I hope somebody someday finds this app useful. If not, well, it sure makes my life easier.

To Download & Install OpenVPN File Manager

1. Make sure "gksu" is installed on your system. Most PCLinuxOS users should already have gksu installed.

2. Put the zip file in your user's home folder and select "extract here". It will be extracted into its own folder, named "OpenVPN File Manager". Make sure the files "OpenVPNFileManager", "Light", and "Dark" inside that folder are marked as being executable and click "OpenVPNFileManager" to start the app.

OpenVPN File Manager currently is NOT in the PCLinuxOS repository. However, you can download it independently, and install it by following these instructions. Click the link below to download the program. Filesize: 15.6 MiB.

DOWNLOAD

Google Stadia: The Good, Bad, & Ugly

by Agent Smith (Alessandro Ebersol)

Google caught everyone by surprise when it announced its Stadia video game platform. Previously, Google's research with video games was known as Project Stream or Yeti.

But despite the surprise that was the announcement, streaming games have been around for some time. Of course now, they are more in vogue than ever.

But before discussing the Stadia, let's look at the history of the so-called Cloud Gaming.

Cloud Gaming, a 19-year history

In 2000, G-cluster demonstrated cloud gaming technology at E3. The original offering was the Wi-Fi cloud gaming service for handheld devices. Video game developer Crytek began researching a cloud gaming system in 2005 for its Crysis game, but halted development in 2007 to wait until infrastructure and cable providers were ready for the task.

OnLive officially launched in March 2010, and its gaming service began in June with the sale of its OnLive microconsole. On April 2, 2015, it was announced that Sony Computer Entertainment had acquired the OnLive patents, and OnLive closed its doors.

In November 2010, SFR launched a commercial cloud computing service in IPTV in France with G-cluster technology. And the following year, Orange France unveiled its gaming service on IPTV based on G-cluster technology.

GeForce NOW is a cloud-based gaming streaming service offered by NVIDIA that was released on October 1, 2015.

Nvidia GRID is a recent creation of Nvidia, focused specifically on gaming in the cloud. Nvidia GRID includes graphics processing and video encoding on a single device that is able to shrink input to display the latency of streaming cloud-based video games. This is important because of the impact that the latency will have between what the user does and when the action is displayed on the screen.

Blade SAS Group launched Shadow, its flagship cloud gaming service in France in November 2011. In October 2018, Shadow announced that it was available in 19 states on the east and west coasts with additional expansion plans throughout the country.

LOUDPLAY announced the expansion of its cloud gaming service to Ukraine, Belarus and a number of other sites in Eastern Europe on May 18, 2018. On November 21, 2018, LOUDPLAY in partnership with Rostelecom and Huawei demonstrated the first showcase of 5G games in Europe (in Innopolis).

Electronic Arts acquired the Gamefly cloud games startup on May 22, 2018. A few months later, on October 29, 2018, Electronic Arts announced the Atlas gaming projects in the cloud.

Google unveiled Project Stream on October 1, 2018. The project was formally announced at the Game Developers Conference on March 19, 2019 as Stadia.

Microsoft introduced the xCloud project on October 8, 2018.

Cloud gaming and streaming also comes to fill the technological gap. Technology walks faster than anyone can keep up with, and with the increase in the price of video cards (thanks to cryptocurrency mining), better computers and more powerful boards are assets that have gone further on the horizon for many game fans.

So we've been looking for a solution for cloud games for the last 19 years. Now, we finally have a solution, in which a company like Google will put all its infrastructure strength to make it a success.

Stadia: The Good!

Imagine being able to play recent games, the so-called AAA, on any device that runs Google Chrome?

You do not need to have the Windows operating system (which is preferred for releasing such games), you do not have to have a super-powerful machine, since the game will be transmitted in images to the remote device, which will be in your house, and you can seamlessly network with your

friends as the game will run on a Google render farm in a datacenter around the world.

But the thing goes further: Making videos of gameplays is quite common and successful on YouTube. With Stadia, you'll be able to make live gameplays, on the fly at the moment you're playing, and stream without any additional requirements, such as OBS or video capture cards. Your game, in real time, will be broadcast on YouTube.

What's more, while you're playing, if the game is a multiplayer one, you can invite your friends to join your game by simply sending a link to your game session.

How will it work?

Any device with an operating system that can run YouTube may be a Stadia gaming station. And, Google promises that any USB-HID joystick can be used. But for the sake of minimizing latency, Google recommends using its proprietary Wi-Fi joystick, connected directly to Google's data centers, to improve the player's experience.

The Google Stadia Joystick

The advantage of the Stadia joystick is that the service can be accessed by Chromecast and played on a large-screen TV with no joystick connection to Chromecast. The Joystick will be connected via Wi-Fi to Google's data center running the game. The joystick will have chat and other interactive functions accessible by additional buttons on the control.

Another feature of having a joystick connected directly to the game is that it will be possible to jump from any screen to another screen and continue playing in a transparent and immediate way (according to Phil Harrison, head of the project at Google).

According to Google, any device running YouTube could be a Stadia receiver

The Official Announcement

Project Stream was the first sign of interest, announced by Google, in video game products. There have been rumors that Google has been working on a service called Project Yeti since at least 2016.

Google also hired gaming industry executive Phil Harrison, and was seen recruiting developers during industry events in 2018. Project Stream's main differential to previous services such as OnLive, GeForce Now, and PlayStation Now is its ability to run on any desktop Chrome browser, rather than specific gaming platforms. The service uses AMD Radeon graphics hardware.

Google announced the service in October 2018 and shortly thereafter opened invitations to beta testers with access to Assassin's Creed Odyssey. Players could request access and those who reached a minimum Internet speed could run the game on their Chrome browsers. Those who participated received a free copy of the game when the trial period ended.

The games available on the service announcement were Assassin's Creed Odyssey and Doom Eternal.

Google Stadia Stand at Game Developers Conference at the Game Developers Conference

Stadia was formally announced during Google's opening speech of the Game Developers Conference 2019 in March 2019. To support Stadia, Google also announced the formation of Stadia Games and Entertainment, its creative studio for original games, with Jade Raymond as the leader. In addition to developing its own games, Stadia Games and Entertainment will help support the transition from third-party titles to the Stadia service.

Harrison stated that "We are based on Linux, we use the Vulkan Graphics API, the developer develops in our cloud instance, so the development kits are now in the cloud. In our cloud, in the private data center of the developer or on his desktop ", affirming the ease and availability of the system for the developer.

One good thing that might come out of it could be the improvement of Radeon Linux drivers, which have never been as good as Nvidia's.

The Not so good

So far, there is no Google definition on how the service will work. And, especially, what will be the charge for services. Will it be charged per game, like Steam? Or will it be charged like Netflix? A monthly fee and the possibility of the subscriber to play the entire catalog of the service?

Other things to consider is that, in the event of service announcement, all screens running Assassin's Creed Odyssey were in a controlled environment, almost without lag or latency. How will it work in real life? With real lag problems and latency? And, will it be dependent on fiber optics?

Of course Phil Harrison ensures that Google is in a position to offer the service by having a large data center infrastructure around the world.

Even then, there were moments, in the demonstration, where the quality of the video was reduced, to keep up the frame rate, and the famous artifacts appeared on the screens.
But, there are other outstanding issues, how about the data cap of internet plans? Yes, depending on the data plan that is signed with the internet provider, your data limit on the plan will be spent quickly with this service (not considering Netflix, You Tube, Hulu and others). So, will Google subsidize the plans?

The Bad

The announcement could not have come at a worse time: The gaming industry is going through a very ugly crisis (not like 1983), with many layoffs and job closures.

At the end of February this year the digital store GOG discreetly dismissed what it claims to be a dozen employees. GOG, which is owned by The Witcher 3 publisher, CD Projekt, did not say why the layoffs happened, but a fired official told Kotaku that the store was in financial trouble.

In the same month of February, Activision Blizzard fired 8% of its workforce, or 800 people. In the announcement of results, Activision Blizzard CEO Bobby Kotick told investors that the company "once again achieved record results in 2018" but that the company would be consolidating and restructuring because of the frustrated expectations in 2018. It would be cutting primarily non-game development departments and reinforcing its development team for franchises such as Call of Duty and Diablo.

ArenaNet, the studio behind the popular online games Guild Wars and Guild Wars 2, also informed its employees that they are planning major layoffs, according to a person who is working there.

So in the midst of a crisis with large digital entertainment companies, the emergence of a new video game platform may be more detrimental than beneficial, after all with the massive use of datacenters, many stages of the electronic gaming will be burned. Thinking in terms of chip factories and electronic boards, many will become superfluous.

As for Google, the more vertical the better, since the company will not use third-party resources, but only use of its own resources.

But why is the video game industry in crisis?

Well, one can speculate on what happens to the gaming industry nowadays.

Over the years, games have become bigger than Hollywood movies. Yes, far gone are the days when solitary programmers created video game games and those were a success. River Raid and Carol Shaw were in the distant year of 1982.

River Raid was a success of Activision in the Atari 2600, created by a single programmer, Carol Shaw

With the evolution of both software and hardware, the games started to require a production with budgets larger than most Hollywood movies. Motion capture artists, voice acting actors, screenwriters, directors, makeup staff, visual effects, sound editing, in short, teams with more than 20 people for super productions that leave many indie movies with envy. Uncharted 4 cost around $ 50 million, excluding advertising spendings, for a team of 150 professionals.

Now, the gaming industry is mirroring itself in filmmakers and betting on franchises: Games can be expensive to produce, so why risk it? If there is a franchise, which the public recognizes and has loyal fans, every year is produced more of the same and everyone is satisfied: fans with games and producers with the return of their investment with profits.

It would be great if it were not the franchise fatigue...

Yes, today, this problem does not only affect movies. As games are heading towards the franchises, the

franchise fatigue is a reality. Today we see 4, 5, 6 titles about the same characters and stories, to the point that fans begin to lose interest. Alien Covenant was a great example of this, a franchise piece that fans have stopped caring about.

As for games, Assassin's Creed is already in its 12th version, the aforementioned Assassin's Creed Odyssey and the question that remains is: Are there so many stories to be told?

Games in franchises have already suffered with fatigue: Need For Speed and Tony Hawk's Pro Skater are two good examples of games that were exploited to exhaustion, and so much so, that today were forgotten.

The Ugly

Well, what I'm going to write now is speculation, but from the observation of previous events, it might happen.

Television as mass media no longer holds up, and people are seeking experiences that are closer to them: You Tube, Vimeo, and so many other personal video services where people create content for people. I myself follow the journalist Glen Grenwald of the Intercept website, and I think he has more credibility than many traditional TV anchors.

With the advent of these communication channels, and especially the measurement of the responses of the people who seek them, it is possible to outline the users, and to know their preferences, their tastes, who are their gaming partners, with whom they relate and etc. ...
And, all profiled.

With all this information in hand, what could happen? What manipulations could be done? The affair Cambridge Analytica / Facebook is there to show us how far people's manipulation can go based on their personal data. Knowledge is power, and knowing too much about an entire population means being able to manipulate the whole population.

Google's motto, "Don't be Evil," has already been forgotten, and has been deleted from its code of conduct when the company underwent a restructuring in 2015. Now we have only to wait when and how Google will be evil, but its collaborations with the Chinese and Pakistani governments show it is already conniving with evil.

The conclusion

No matter how cool the aspects of cloud gaming and streaming, a basic thing will die: Players will no longer own their games. And because of that, they will no longer be able to play whenever they want, but when the cloud gaming company makes the games available. There will no longer be possession, but only a glimpse, a look of what could be of the player, but for N reasons is not, nor will be.

They will no longer own the video games consoles, after all, the whole act of playing will be abstracted, and the player will have only the visual and sound experiences, that may degrade, depending on the conditions of the network and data centers.

But for now, we can calm down and be momentarily relieved: Google Stadia, as it stands, is just vaporware, and depending on how the infrastructure evolves (or doesn't), it will be another Google product dead on arrival, such as Google glasses, an interesting technology that never took off.

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.

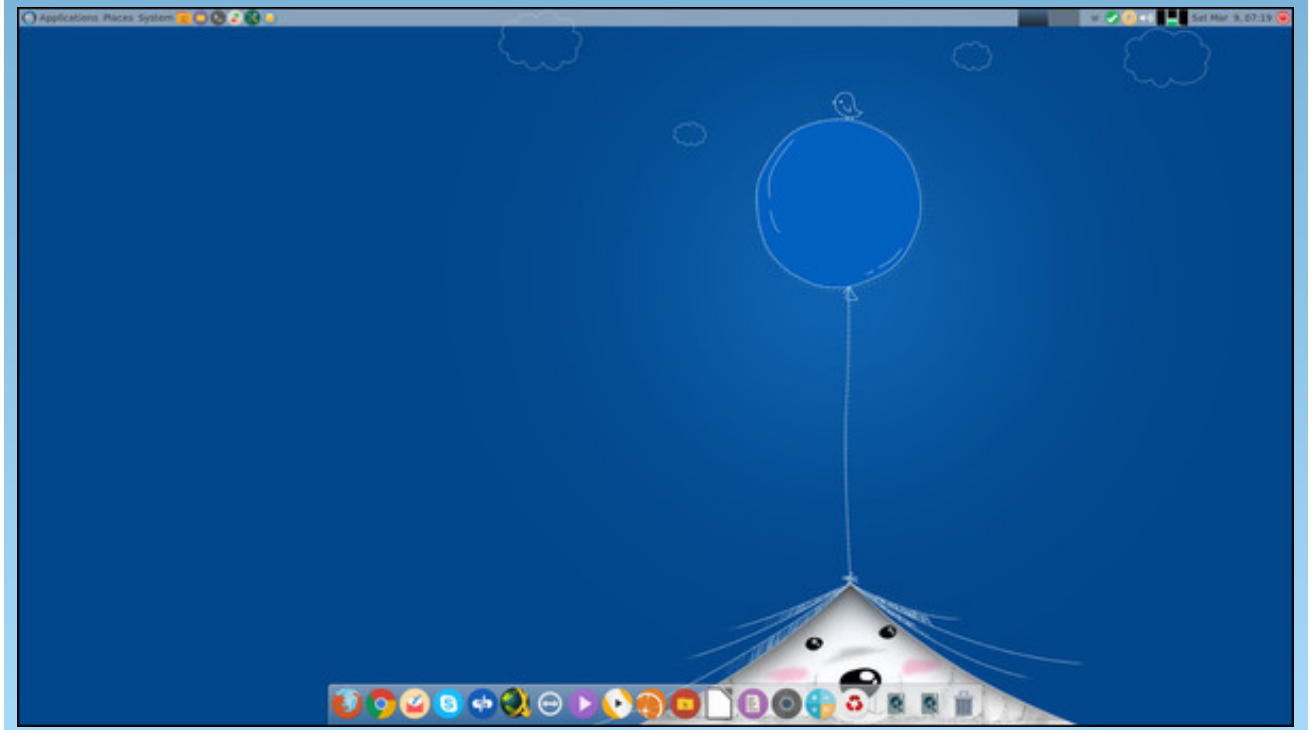
4. No representations, warranties or guarantees whatsoever are made as to the accuracy, adequacy, reliability, completeness, suitability, or applicability of the information to a particular situation. All trademarks are the property of their respective owners.

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 Users
A majority of sections in the magazine contain materials submitted by users. The PCLinuxOS Magazine accepts no responsibility for the content, accuracy, conformity to applicable laws of such material.

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

Screenshot Showcase



Posted by jogurtmen, March 9, 2019, running Mate.

PCLinuxOS Family Member Spotlight: oldfrt

As told by YouCanToo

What is your name/username?
Ron Campbell / oldfrt

How old are you?
74

Are you married, single?
Married

How about Kids, Grandkids (names and ages)?
Just one; he is four (4).

Do you have pets, what is your favorite?
We have a cat, which we rescued from the Niagara Cat Adoption.

Are you retired, still working and if working, what do you do?
I am retired. I worked for IBM for 47 years in large mainframes. I did not work in PC's. My expertise was in what is called zVSE today.

Where do you call home? What is it like? IE: weather, scenery
We currently live in Niagara Falls, Ontario, Canada. I grew up in a suburb of Toronto called East York, where I went to Public School then High School.

When retirement was coming close we decided to get out of the big city and move down the road to Niagara Falls. The Falls is now home to a lot of retirees who enjoy a much quieter and easy lifestyle.

We are about five minutes from the tourist area, but then again, everything is about five minutes away in the Falls. It's 120 km to Toronto, and about 40 km to Buffalo, New York. Niagara Falls is situated in the 'banana belt,' as it is affectionately called. If you look at a map of Lake Ontario, there is a curve in the west end of the lake going from Toronto, Hamilton, then Niagara Falls area shaped like a banana. Our seasons tend to be milder than it is in Toronto, due to the Lake effect, which keeps us warmer in the summer but not as cold as Toronto in the winter. In the winter, the winds come across Lake Erie and then dump all the snow on Buffalo. So, we usually do not get as much snow as our American friends, but we still love them anyway.

Where did you go to school and what is your education level?
I graduated from Ryerson Polytechnical Institute, Toronto in 1967. At that time, Ryerson graduated a lot of technical people. While we were not at the level of University engineers, Ryerson skills where highly coveted by many companies. Today, Ryerson is a full University and has a high reputation for producing competitive grads. I went right from Ryerson to IBM.

What kind of things you like doing? hobbies, travel, fishing, camping?
I build models. All kinds of different models, R/C boats, R/C airplanes, plastic models, static wooden boats. During the summer, I try to get out and participate in flying my model planes. During the winter, I fly indoor R/C planes at an indoor soccer facility in Welland which is about 15 minutes away.

I also have a 1973 Oldsmobile Cutlass Supreme, of which I am the original owner. I spent about 10 years restoring it, and we now take it to cruise nights throughout the Niagara Region. I used to have a web site provided by my cable company where I featured my car, but the cable company decided that it wasn't worth the effort, so they pulled the plug on user web sites. Otherwise, I could show you my car.

Why and when did you start using Linux?
My very first computer was a Timex 1000 with 2k of RAM. I spent a lot of time coding programs and putting them on tape. It was a lot of fun. Eventually, I bought a 286, and then progressed through the years working my way up from DOS to Windows and various used computers that other people were throwing out. I was very cheap in those days. Well, I guess I still am. At work, I started using a green screen dumb terminal. By the time I retired, I was using a laptop running Windows up until my last year, when I was on a pilot project using Linux on a laptop. I supported the mainframe systems at IBM, what is known today as z/OS, z/VSE, z/VM and also Linux running on these mainframes.

Around 2000, IBM was promoting Linux and I wanted to learn how it worked. There were many free versions that I came across, and I tried many of them with limited success. It wasn't until I found a magazine with a free copy of a new distribution called PCLinuxOS that I actually got something that I could use. I have been a user ever since. At work, I messed around with Linux Servers running under z/VM, but they were not desktop systems. PCLinuxOS was my at home workhorse.

What specific equipment do currently use with PCLinuxOS?
I have always used something that came to me second hand, or that I rescued from the dump. Last year, I finally decided that I wanted something better and blew my allowance on a custom built cpu. There is a young gentleman down the street from me who has a computer store, so I asked if he could build me something. I am not a gamer, so I just needed something current and reliable. He put together a nice system which did not break the bank, and I am quite happy with it.

Do you feel that your use of Linux influences the reactions you receive from your computer peers or family? If so, how?
I like to give folks a hard time about using Windows, but it's just for fun. People think that because I worked for IBM, I must be some kind of PC expert, so I do get a lot of questions. I have to be patient and explain that I am not a PC expert, but sometimes they just don't get it. I like to help people, so I try as best I can to assist where necessary.

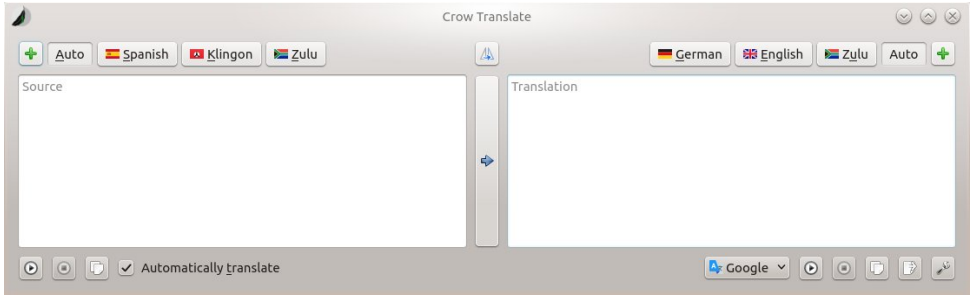
What would you like to see happen within PCLinuxOS that would make it a better place. What are your feelings?
Things are great, the forum is fabulous, and there are a lot of fun people there. You can even find a ton of help. The only thing I have to remind people of is that not everyone is an expert, and when you help people, you have to assume they know nothing. Far too many times I have seen an answer, but no explanation. The answer 'turn option 2 to off' may work, but I think it is better to stand back and explain what effects the options are, and why we change it to something else and what that does. Yes, it takes a few more minutes, but nobody started out by knowing everything. I spent most of my career in customer support, and established a reputation for explaining the problem and solutions better than others.

PCLinuxOS Family Member Spotlight is an exclusive, monthly column by YouCanToo, featuring PCLinuxOS forum member. This column will allow "the rest of us" to get to know our forum family members better, and will give those featured an opportunity to share their PCLinuxOS story with the rest of the world.
If you would like to be featured in PCLinuxOS Family Member Spotlight, please send a private message to youcantoo, parnote or Meemaw in the PCLinuxOS forum expressing your interest.

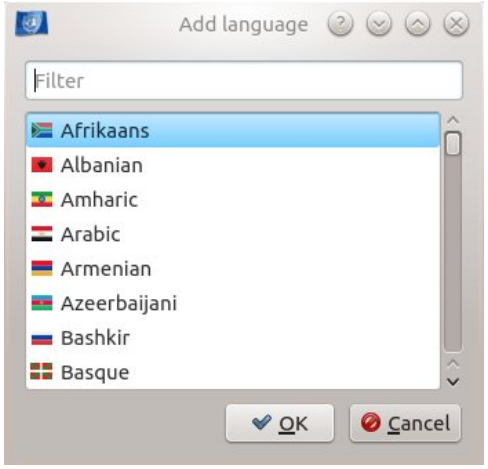
Repo Review: Crow Translate

by CgBoy

Crow Translate is a simple application for translating text into 117 different languages, using Google, Bing, or Yandex online translation APIs. It has both a graphical user interface, and a command-line interface. When you launch Crow Translate, it will appear minimized in the system tray. Just click on the icon to open the main window.



On the left side is the source translation box, where you'll put the text you want to translate, and on the right is the output. By default, it automatically translates everything you type in it. Above those are the language selection buttons. To add a language, simply click on the green + button. You can only add three languages at a time. In the lower right is a drop down menu where you can select which online translation API to be used.



Another thing you can do is just select text in any program and press **Ctrl+Alt+E** to automatically bring up a small Crow Translate window, with the selected text pasted in and translated.

Crow Translate can also play back the translated text using text-to-speech engines provided by the translation API used (Google, Yandex, or Bing). The playback buttons are underneath the translation text boxes.

I unfortunately ran out of time to really test out Crow Translate's command-line interface. But you can just type **crow -h** into a terminal to see the various options and usage.

Summary

I found Crow Translate to work pretty well for the most part. The interface was slightly confusing at first, and at times Google's translation API did give errors (Probably Google's fault, not Crow's). But overall I found it to be a nice alternative to the web based translation sites. However, it still can't fully translate web pages, unlike the web based Google Translate.

The Ruby Programming Language: Writing A Ruby Program

by phorneker

Last time, I introduced you to the Ruby Programming Language by way of the Interactive Ruby interpreter. Towards the end of the article, I gave you a traditional "Hello World" program and introduced you to class implementation. That took a fair amount of typing to accomplish what we did. Now, instead of having to type that code in Interactive Ruby every time, we can create a source code file in the Ruby language using any text editor available in the PCLinuxOS repository, be it kate, gedit, or even EMACS, the latter of which I prefer to use for writing source code that is not going to be part of a website. (For HTML, I use Bluefish.)

Let us take that classic "Hello World" program we wrote in the last issue.

```
def hi(name = "World")
puts "Hello #{name.capitalize}"
end
```

Instead of having to type in all of this into irb every time we launch irb, would it not be easier to simply type this program into an editor and save it as **helloworld.rb**? That is exactly what we are going to do.

Ruby source files have **.rb** for a file extension, so there is no doubt that this is a source code file that was written in Ruby.

If you are using a desktop such as Plasma 5, XFCE, MATE, or even LXDE, double clicking on an icon representing that file will associate the file with the installed Ruby interpreter (unless you changed the association to do something else, such as assign the Ruby source file type to a text editor for editing that file).

First, we launch a text editor.

All Ruby source files start with the following line to indicate that this is a source code file written in Ruby.

```
#!/usr/bin/ruby
```

This is the location where the Ruby interpreter binary is located after installing the ruby package from the repository.

I have seen this statement written as **#!/usr/local/bin/ruby** on systems where Ruby was compiled from source code using the standard configure, make and then make install method associated with compiling from source code tarball packages. But, for PCLinuxOS, if you installed the ruby package from the repository, the ruby binary will always be located at **/usr/bin/ruby**.

Now, let us save the file.

For this article, I created a ruby directory within the home directory on my laptop. So, the Hello World program is stored at /home/patrick/ruby/helloworld.rb, which now contains the following:

```
#!/usr/bin/ruby

def hi (name = "World")
puts "Hello, #{name.capitalize}"
end
```

The hashtag (#) at the beginning of any line is used to comment or otherwise make notations on the source code so that others who read this source code will know what is going on in this particular program, function, method, class or even a entire Ruby library.

The first line is a notable exception to this rule. The **#!** indicates that the remainder of this line points to the location where the interpreter or shell binary is located, such as the Ruby interpreter, followed by any parameters passed to the interpreter or shell.

```
#!/usr/bin/ruby

def hi (name = "World")
puts "Hello, #{name.capitalize}"
end
```

There is a (optional) parameter that is passed to the function hi called name and has been assigned the string value of "World". This value is used only if the function hi is called without any parameters passed to that function.

The #{name.capitalize} within the quotes in the puts statement is a placeholder that first, makes sure that name is properly capitalized before displaying the contents of name.

As everything in Ruby is an object, variables are no exception, and task of the capitalize method of the name variable (as an object) is to make sure that the value assigned to name is properly capitalized. (Yes, it is that simple.) The capitalize method is one method used by Ruby for its string handling functions.

This program is not yet complete. We need to invoke this function, with a statement called (what else?) **hi**.

I have included two versions of this statement to show you the two ways this function can be called.

In the second version, be sure to place quotes between the name, or you will get a runtime error when you run this program.

```
#!/usr/bin/ruby

def hi (name = "World")
puts "Hello, #{name.capitalize}"
end

hi()
hi("Patrick")
```

Now save the file and exit your editor. We can now open a terminal and look at the file(s) we created. Simply type ***ruby helloworld.rb*** and press the Return/Enter key.

```
[patrick@localhost ruby]$ ruby helloworld.rb
Hello, World
Hello, Patrick
[patrick@localhost ruby]$
```

We have just successfully created the Hello World program in Ruby. The first version of the hi function was called without any parameters. Since the default value for name was assigned the value "World", that is what was used for the parameter. As this was a script that ran on the Ruby interpreter instead of Interactive Ruby, we did not see the "**=> nil**" that would normally appear.

Bonus Tip: If you wish to execute the script without having to explicitly type "ruby" on the command line, simply type ***chmod a+x helloworld.rb*** on that command line. You will, however need to type ***./helloworld.rb*** to execute the script.

Another Bonus Tip: If you have a **bin** folder in your home directory, move this file to that directory, then all you have to do is type ***helloworld.rb*** to execute the script.

A COMMENT ON COMMENTS

I mentioned that the hashtag (#) is used for placing comments in Ruby source code. This is only one way to accomplish this task. Comments with a hashtag at the beginning of the line are ignored by Ruby for that particular line of Ruby code. Placing a hashtag at the end of a function or statement in Ruby causes Ruby to treat the remainder of that particular line of code as a comment at the same time interpreting the Ruby code that preceded the hashtag, unless that hashtag is inside quotes in a statement such as ***puts***.

In that case, the hashtag is part of a placeholder containing the name of the variable to be used (as well as any methods that modify the variable) when interpreting what is in the quotes. In the case of puts, the placeholder substitutes the value contained in the variable for the name of the variable. For example:

```
puts "Hello, #{name.capitalize}"
```

In this example, the value contained in the variable name is substituted for the placeholder. Had this example read puts "Hello, {name.capitalize}", we would have gotten something we did not expect. Suppose we appended another hashtag at the end of this statement.

```
puts "Hello, #{name.capitalize}" #
This is a Hello World program
```

Ruby would interpret the statement the same way as before. For the second hashtag, the words **"This is a Hello World program"** are treated as a comment, and hence, are ignored by the interpreter.

Now, what is we want to place internal documentation, i.e. comments that take up more than one line of text in the source code. We could start each line with a hashtag...or we could format the comments this way:

```
= begin

Comments placed here, as many lines as you
like

= end
```

...which as we can see is far more readable for us, especially when we review the code sometime in the future.

In Ruby, everything between the "**= *begin***" and the "**= *end***" is interpreted as program comments (aka documentation), and hence is ignored by the interpreter.

VARIABLES ARE OBJECTS, TOO.

In Pascal, variables are explicitly declared as to its scope and data type before they are used in a program, a procedure or a function. In Ruby, however, variables are objects, that is, they are nothing more than data and methods to manipulate that data.

Let us examine this code fragment again.

```
def hi (name = "World")
puts "Hello, #{name.capitalize}"
end
```

In the function **hi**, the parameter name becomes optional rather than required as a default value is supplied if no parameter is passed to the function at the time of invocation.

So what is **name.capitalize**? An object consisting of a variable called name, which incorporates the capitalize, uppercase, and lowercase string processing methods (or functions if you will) associated with that object (presuming, of course, that name contains a string of characters).

For example, if ***name = "world"*** then ***puts "Hello, #{name.capitalize}"*** would output

```
Hello, World
```

Likewise, ***puts "Hello, #{name.uppercase}"*** would output

```
Hello, WORLD
```

and as we would expect, ***puts "Hello, #{name.lowercase}"*** would output

```
Hello, world
```

This would be very difficult if not impossible to implement in a traditional language such as Pascal, which was designed for structured programming, rather than object oriented programming. (Delphi and Turbo Pascal 7.0 are variants of Pascal that implement object oriented features, but Pascal itself was not intended to be a OOP language.)

Better yet, try doing this exercise in BASIC!

Technically, Ruby has only objects and methods, and they are created and deleted on demand whereas Pascal has variables, data types, procedures, and functions, all of which must be explicitly declared before they can be used in a program.

If you are looking for data types in Ruby, they do not exist, at least in the traditional sense. Variables do not need to be explicitly defined before they are used.

However, if you wish to "declare" variables (in the tradition of Pascal) in a Ruby program, I recommend documenting these variables and their intended types by placing these declarations in a comments section of the source code. The data types used by Ruby are classified as follows:

Strings: Variables created and used here consist of zero or more characters enclosed in quotes, or simply entered from a keyboard, a disk file, or other input stream. Anything that makes up a string in the C, C++ and Java languages is also a string in Ruby. This includes the "***/n***" (for new line, or ASCII code ***0x0a***), "***/r***" (for the return carriage, or ASCII code ***0x0d***), and "***/t***" (for tabulate, or ASCII code ***0x09***).

Numbers: Ruby supports integer as well as floating point numbers, just as with any traditional language.

INPUT AND OUTPUT STATEMENTS

In our Hello World example, we used the puts function to output objects. Normally, this statement outputs to what we call stdout in C and C++, or the standard output device, usually the screen or terminal window. From the command line, this can be redirected to a disk file, a network (TCP/IP or UDP/IP) port, a UNIX pipe, or a print queue (through the lpr command).

Just as puts produces output from objects, we can also allow Ruby to ask for input of data to objects. The gets function does just that. There are some differences on what parameters gets supports.

First, a variable object is required as a parameter.

Second, ***puts*** allows placeholders to be included within quoted strings when accompanied by their corresponding variable objects. One would think that including such a string would display the quoted string as a prompt before asking for input. But, this is not the case.

As ***gets*** function was intended for input of data to variable objects, it simply would not make sense to assign data input to quoted strings, including those that include placeholders.

So, how do we get a prompt on the screen for input? We could use ***puts***, but puts automatically places the cursor at the beginning of the next line on the screen before asking for input. (For the PCLinuxOS implementation, this is done with a return carriage character.)

Ruby implements the ***print*** function that does the same thing as ***puts*** with one exception, i.e. the cursor is kept on the same line as the object that was just output. So, how would this work in our Hello World program?

We could have written the Hello World program as follows:

```
#!/usr/bin/ruby
puts "Hello World"
```

...and we would have been done with it. (This particular example happens to be the world's shortest "Hello World" program ever, with Ruby taking the honors for this implementation.)

Having said that, we could implement a way to allow the input of a name, then use that name to output the greeting.

```
#!/usr/bin/ruby

print "Please type in your name "
name = gets
puts "Hello, #{name}"
```

As gets is a function, called without parameters, it will always return a string object.

WE GOT YOUR NUMBER

In this example, if you type in a number, gets returns that number as a string containing the characters that represent the number.

For instance, if you type **1048576**, gets assigns "**1048576**" to the object name. This object can be converted to a floating point number and assigned to another variable object.

```
#!/usr/bin/ruby
print "Please type in your name "
name = gets
number = name.to_f
puts "Hello, #{name}"
```

The method ***.to_f*** converts the contents of the variable object from which ***.to_f*** was called (in this case, name) from a string of characters to a floating point number. If the conversion is successful, the floating point number **1048576.0** should be assigned to number.

If the conversion of that same number is **not successful**, then number would be assigned a floating point value of **0.0** (i.e. zero point zero, or the grade point average of Bluto Blutowsky in the movie *Animal House*.)

Of course, we would not know whether this was successful at this moment until we output number using either **print** or **puts**.

Suppose we wanted to enter an integer instead of a floating point number. The following code would accomplish that:

```
#!/usr/bin/ruby
print "Please type in your name "
name = gets
number = name.to_i
puts "Hello, #{name}"
```

As we can guess, **.to_i** converts string objects to integers the same as **.to_f** converts strings to floating point numbers.

Placeholders used in the **print** and **puts** statements work the same for floating point numbers and integers as it does for string objects, so we can then write the following statement:

```
puts "You have typed in the number #{number}"
```

In place of or appended to the previous puts statement. Let us incorporate this into our hello world program, and save this as **numbertest.rb**.

```
#!/usr/bin/ruby

print "Please type in a number "
name = gets
number = name.to_f
puts "You have typed in the number #{number}"
```

Now, let us execute this twice. Once with a real number, and once with random garbage.

```
[patrick@localhost ruby]$ ruby numbertest.rb
Please type in a number 1048576
You have typed in the number 1048576.0
[patrick@localhost ruby]$ ruby numbertest.rb
Please type in a number lame duck
You have typed in the number 0.0
[patrick@localhost ruby]$
```

The first run of **numbertext.rb** came out as we expected. But, look what happened when we entered "**lame duck**" for a number. Ruby was able to tell that "**lame duck**" is not a number and assigned the value **0.0** to number for output.

If we were to implement this program in a traditional language such as Pascal, Fortran, C, or even Python, entering this type of input where a floating point or integer is expected would have resulted in a runtime error and the program would have terminated at the point where the input was asked.

NUMERIC AND LOGICAL OPERATORS

As with any programming language, there are numeric operators (in addition to the functions supplied with Ruby's Math module)

Exponents: When we enter a number such as 2^8 into a LibreOffice Calc spreadsheet, we would type in **2^8**. In Ruby, that same expression is typed in as **2**8**, in both cases, the result would be **256**.

Unary logical operators: In mathematics, when we want to express the negation of a logical expression or variable, such as **not X**, we would write **!X**. Ruby implements the not logical operator the same way we write it in a logical equation. If we wanted to find the **logical complement of X**, we would type **~X**.

Likewise, the unary operators + and - work in Ruby the same way we type them out in mathematical equations on paper or by pressing the +/- key on a calculator, with one exception: when used as a method, + and - are typed in the Ruby code as +@ and -@ respectively.

Basic mathematical operators: The standard symbols we use to code mathematical equations for addition (+), subtraction (-), multiplication (*), and division (/) work the same in Ruby as it does in most other languages and spreadsheets. The modulus (%) works the same way in Ruby as it does in C and C++. (This makes sense as Ruby was written in C to begin with.)

Bit shifting operators: Those of you who are familiar with C and C++, should be familiar with the **<<** and **>>** operators. For those of you who are not, the << and >> operators take the number to the left of the operator and shift the bits in that number left (<<) or right (>>) the number of bits specified to the right of the operator.

Think of **(x << y)** as **(x * (2**y))**, and **(x >> y)** as **(x / (2**y))**.

Bitwise logical operators: The symbol "**&**" performs a **bitwise AND** on two numbers that are supplied to this operator. For instance (**0xFF & 0x08**) will yield **8**, which also happens to be the hexidecimal number **0x08**.

Likewise, the symbol "**|**" performs a **logical OR** on two numbers that are supplied to this operator.

One would think that the "^" operator would represent "to the power of" in an equation, especially when entering exponents on a LibreCalc spreadsheet. But this is not the case. The "^" operator performs a **Exclusive OR** on two numbers that are supplied to this operator.

(Because of this, the "**" is defined to mean "to the power of" instead of "^", which could cause confusion to programmers and machines alike.)

What is the difference between OR and Exclusive OR?

The logical operator **OR** will return true *if one or both parameters in the equation is true, and false if none of the parameters in the equation is true.*

However, the logical operator **Exclusive OR** (sometimes written as **XOR**) will return true *if one or the other parameter in the equation is true, **but not both.*** **XOR** will return false if both parameters are true or both parameters are false.

To show it another way:

```
( true | true ) = true
( true | false ) = true
( false | true ) = true
( false | false ) = false

( true ^ true ) = false
( true ^ false ) = true
( false ^ true ) = true
( false ^ false ) = false
```

Comparison operators: In Ruby, the comparison operators work as expected, namely:

<=, <, >= and **>**

Logical operators: The "&&" and "||" represent the AND and OR operators repectfully, but they perform logical comparisons instead of bitwise operations on numbers. Also, these operators are used for non-number object comparisons as well.

The "==" stands for "is equal to", or "is the same as" when it comes to logical comparisons. Likewise, "!=" stands for "not equal to" or "not the same as" in a similar manner.

STRING MANIPULATION WITH NUMERIC OPERATORS

When does 5 x 3 = 555? When the 5 is in quotes. The "**+**" and "*****" operators normally used in mathematical equations also apply to string manipulation in Ruby.

The "+" operator, when used for string manipulation, concatenates two (or more) strings. For example, we can assign a variable object name as follows:

```
name = "PC" + "Linux" + "OS"
```

When this statement is executed by the Ruby interpreter, name will contain the value "**PCLinuxOS**" (I would hope so)

Likewise, the "*****" operator does the following:

```
name = <string> * <number of times to replicate that string>
```

For example,

```
puts 5 * 3
```

would give us a value of **15**, whereas

```
puts "5" * 3
```

Would give us a value of **555**. In this instance, the number **5** is treated as a character which happens to be "**5**", hence the value returned by "*****" as a string manipulator is the original "**5**" appended by two copies of "**5**".

If we has written

```
puts "5" * 3
```

as

```
puts 5 * "3",
```

then puts would have displayed **33333**.

However, if we had written

```
puts "5" * 3
```

as

```
puts "5" * "3",
```

we would have gotten an error message indicating that one of those parameters must be an integer (or otherwise be able to be converted to an integer) for the "*" operator to work for string manipulation.

On the other hand,

```
puts "5" + "3"
```

would result in the display of **53** on the screen.

IF/THEN CONTROL STATEMENTS

Let us revisit the **numbertest.rb** program.

```
#!/usr/bin/ruby

print "Please type in a number "
name = gets
number = name.to_f
puts "You have typed in the number
#{number}"
```

When I typed "**lame duck**" where it said "*Please type in a number*", the program responded with "*You have typed in the number 0.0*".

The same program would respond with "*You have typed in the number **0.0***" if I had actually typed in either "**0**" or with "**0.0**", and as it is, there would be no way to distinguish between "0" and random garbage, as both would return the number zero ("**0.0**").

There is an easy fix for this problem.

Like any well designed programming language, Ruby comes with statements that control the flow of the program. The easy fix here is to insert a **if/then/else** statement that tests the input we entered to see if we really typed in zero ("0") or if we typed in random garbage.

In Ruby, if/then/else statements are coded as follows:

```
if <condition> then
<block of code to be executed if the test passes>
end
```

We can include code to be executed if the test does not pass.

```
if <condition> then
<block of code to be executed if the test passes>
else
<block of code to be executed if the test does not pass>
end
```

The keyword **end** here is *required* to mark the end of the if/then statement.

In our **numbertest.rb** program, we can code the if/then statement as follows:

```
#!/usr/bin/ruby

print "Please type in a number "
name = gets
number = name.to_f
if (number != 0.0) then
validnumber = true
else
if (name == "0.0") || (name == "0") then
validnumber = true
else
validnumber = false
end
end
if validnumber then
puts "You have typed in the number #{number}"
else
puts "Sorry, what you typed in is not a valid number"
end
```

Now, save this file and run the tests again.

```
[patrick@localhost ruby]$ ruby numbertest.rb
Please type in a number 0.0
You have typed in the number 0.0
[patrick@localhost ruby]$ ruby numbertest.rb
Please type in a number lame duck
Sorry, what you typed in is not a valid number
[patrick@localhost ruby]$ ruby numbertest.rb
Please type in a number 0
You have typed in the number 0.0
```

What did we do? We created this block of Ruby code to test the input we typed in.

```
if (number != 0.0) then
validnumber = true
else
if (name == "0.0") || (name == "0") then
validnumber = true
else
validnumber = false
end
end
```

The variable **validnumber**, while not explicitly typed, is considered to be a **boolean** variable, i.e. a variable that can have only **true** or **false** for valid values.

*In Ruby, the value **nil** is considered to have the value of **false** in logical tests. The difference between **false** and **nil** is returned from any function where no result is **expected**, such as a **puts** statement.*

The first thing we checked in this block of code is to see if the number converted from our input is anything other than zero. If it is, then **this has to be a valid number** as the **.to_f** and **.to_i** methods for number conversion return zero if a string that does not make up a number is passed as a parameter.

If/then/else statements can be nested as I have shown you here. This nesting is necessary for the next test to be included in this block of code.

The next test shown here contains a **logical OR** statement. The number zero can be entered as either "**0**" or as "**0.0**". (There are more ways to enter the number zero, but this is sufficient to demonstrate the if/then/else statements.)

If this test passes, then we can say that what we entered is indeed zero, and not random garbage.

What about the other block of code?

```
if validnumber then
puts "You have typed in the number #{number}"
else
puts "Sorry, what you typed in is not a valid number"
end
```

This simple if/then/else statement simply outputs the appropriate message depending on whether we entered a valid number or not.

Let us go back to the other block of code:

```
if (number != 0.0) then
validnumber = true
else
  if (name == "0.0") || (name == "0") then
    validnumber = true
  else
    validnumber = false
  end
end
```

This could have been coded another way.

```
if (number != 0.0) then
validnumber = true
else
  validnumber = false
  if (name == "0.0") || (name == "0") then
    validnumber = true
  end
end
```

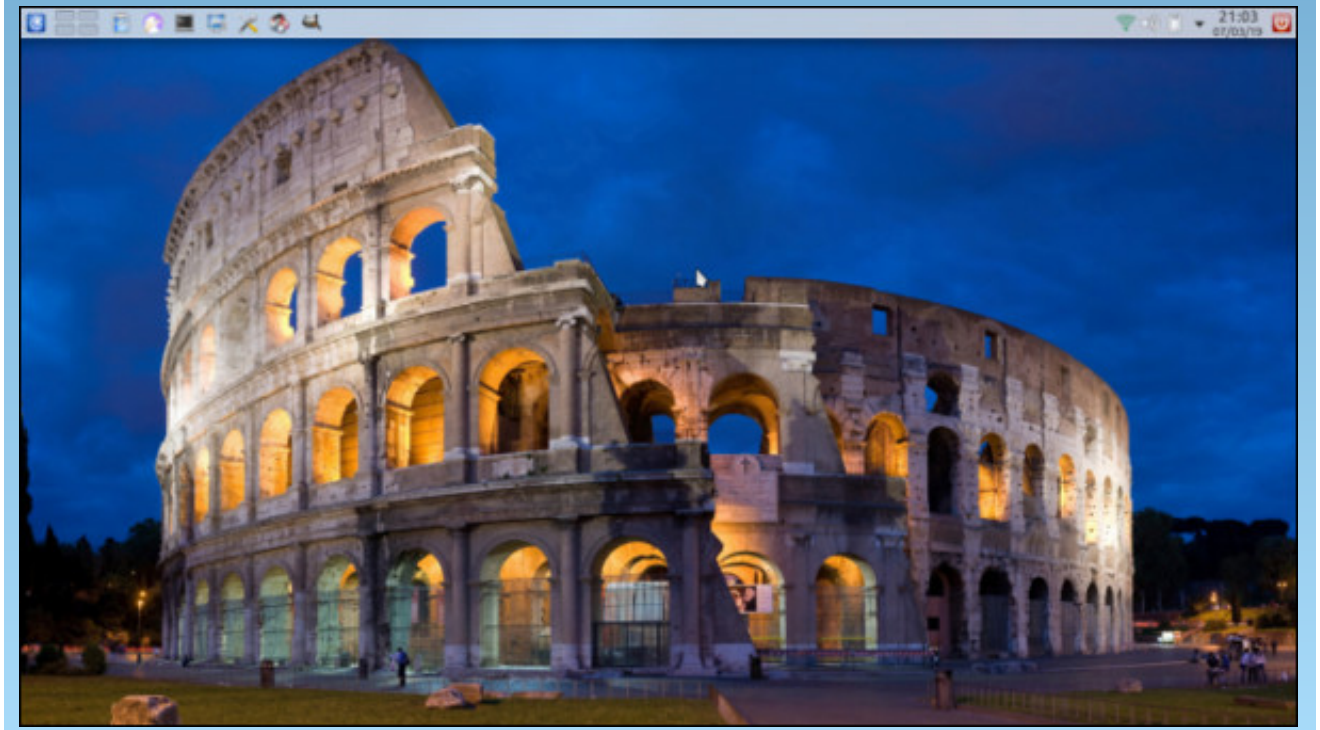
Here, **validnumber** changes *only if* we can prove the input to really be a valid number.
While both of these achieve the same result, I prefer the former as the logical flow is more obvious than the latter. But then, that is just a matter of opinion and style.

WHAT WE ACHIEVED SO FAR

So far, we have been able to write simple Ruby programs with the use of variables and control structures.

We are getting a feel now for what it is like to program in Ruby. Next time, I shall get more into string manipulation and control structures.

Screenshot Showcase



Posted by hurricane, March 7, 2019, running KDE.

Happy 30th Birthday, WWW!

by Paul Arnote (parnote)

Imagine browsing the internet pre-1989. Your search for information encompasses searches of USENET newsgroups, gopher searches, email, archie and veronica searches, telnet, and FTP'ing files. Oh, the information you seek is out there, but the real chore is finding it. There are no search engines like we have now. Only through perseverance, sheer determination, and many hours of work are you able to find the information you seek.

Then, comes along a man with a vision.

"Suppose all the information stored on computers everywhere were linked. Suppose I could program my computer to create a space in which everything could be linked to everything," said Sir Tim Berners-Lee.

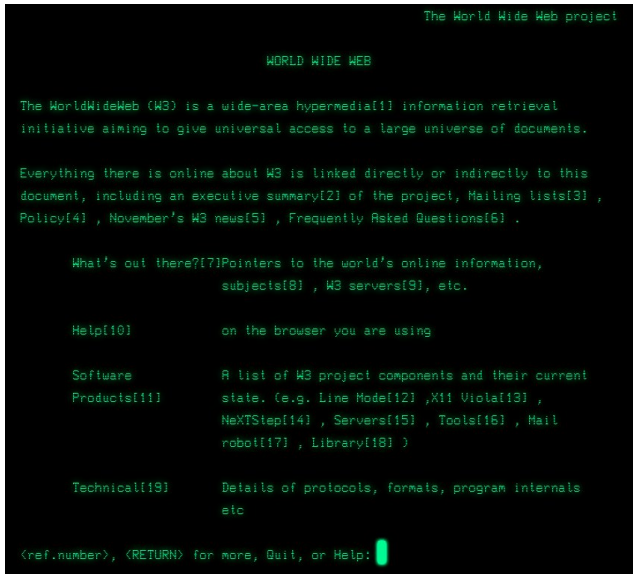
Sir Timothy Berners-Lee arriving at the Guildhall to receive the Honorary Freedom of the City of London September 24, 2014, Paul Brown, CC-SA 4.0

And so, he embarked on a project to do just that.

Finally, in March 1989, Berners-Lee made a proposal while working at CERN (at that time, the largest internet node in Europe) for an information management system that would interlink data on computers on a network via hypertext. Later, in November, 1989, he successfully got a client and server on the internet to talk to one another, using HyperText Transfer Protocol, or otherwise known as HTTP.

To create it, Berners-Lee had to come up with the URI (Uniform Resource Identifier) system that assigned an addressing scheme to find a document, the HTTP protocol to connect the computers on a network together, and the HTML (hypertext markup language) format, which formatted the pages containing the links.

He then had to create a web server ***and*** a web client. It was a tall, tall task, the results of which we all benefit from today. Berners-Lee made it clear then, and continues to evangelize that the world wide web is for everyone.



First web page, still running today.

NeXTStep computer used to develop the WWW. Photo: CERN.

You can still view the first web page to ever grace the world wide web here. The world's first web server was a NeXTStep computer. From Wikipedia:

Mike Sendall (Berners-Lee's manager) buys a NeXT cube for evaluation, and gives it to Tim [Berners-Lee]. Tim's prototype implementation on NeXTStep is made in the space of a few months, thanks to the qualities of the NeXTStep software development system. This prototype offers WYSIWYG browsing/authoring! Current Web browsers used in 'surfing the internet' are mere passive windows, depriving the user of the possibility to contribute. During some sessions in the CERN cafeteria, Tim and I try to find a catching name for the system. I was determined that the name should not yet again be taken from Greek mythology..... Tim proposes 'World-Wide Web'. I like this very much, except that it is difficult to pronounce in French... by Robert Cailliau, 2 November 1995.

Here's some more information on Berners-Lee, courtesy of CERN:

Sir Tim Berners-Lee invented the World Wide Web in 1989. He is a scientist and academic whose visionary and innovative work has transformed almost every aspect of our lives.

Having invented the Web in 1989 while working at CERN and subsequently working to ensure it was made freely available to all, Berners-Lee is now dedicated to enhancing and protecting the web's future. He is a Founding Director of the World Wide Web Foundation, which seeks to ensure the web serves humanity by establishing it as a global public good and a basic right. He is also Director of the World Wide Web Consortium, a global web standards organisation he founded in 1994 to lead the web to its full potential. In 2012 he co-founded the Open Data Institute (ODI) which advocates for Open Data in the UK and globally. Sir Tim has advised a number of governments and corporations on ongoing digital strategies.

Sir Tim has received multiple accolades in recent years. These include receiving the first Queen' Elizabeth Prize for Engineering in 2013, election as a Fellow of the American Academy of Arts and Sciences in 2009 and being knighted by H.M. Queen Elizabeth in 2004. He has received over 10 honorary doctorates, is a member of the Internet Hall of Fame, and was awarded the Finland Millennium Prize in 2004, and the A.M. Turing Award — often called 'computing's Nobel Prize' — in 2016. In 2007, Berners-Lee was awarded the UK's Order of Merit – a personal gift of the monarch limited to just 24 living recipients and was named one of Time Magazine's '100 Most Important People of the 20th Century'. In 2012, he played a starring role in the opening ceremony for the Olympics, where, in front of an audience of some 900 million, he tweeted: "This is for everyone".

Over the past 30 years, the web has seen phenomenal growth. In so many ways, it has changed the very fabric of society. It has made more information available to the masses than anything else that preceded it.

Today, it continues its rapid growth. There are whole generations of people who can't imagine life without the world wide web. For them, it is something that has always just been there. They just don't know the world without it.

Want To Read More About It?

Here are some other resources where you can read more about the founding of the web.

Google Arts & Culture Editorial Feature
The World Wide Web: The Invention That Connected The World

Internet Society
Celebrating the 30th Anniversary of the World Wide Web

World Wide Web Foundation
History of the Web

Google Arts & Culture (Pictorial)
The Birth Of The World Wide Web

Wikipedia
World Wide Web

Like Us On Facebook!
The PCLinuxOS Magazine
PCLinuxOS Fan Club

Screenshot Showcase

Posted by Yankee, March 24, 2019, running Mate.

PCLinuxOS Bonus Recipe Corner

Pasta Primavera

Ingredients

8 ounces uncooked fettuccine or linguine
1 tablespoon olive or vegetable oil
1 cup broccoli flowerets
1 cup cauliflowerets
2 medium carrots, thinly sliced (1 cup)
1 cup frozen green peas, rinsed to separate
1 small onion, chopped (1/4 cup)
1 container (10 ounces) Alfredo pasta sauce
1 tablespoon grated Parmesan cheese

Directions

1. Cook fettuccine as directed on package.

2. While fettuccine is cooking, heat oil in 12-inch skillet over medium-high heat. Cook broccoli, cauliflowerets, carrots, peas and onion in oil 6 to 8 minutes, stirring frequently, until vegetables are crisp-tender.

3. Stir in Alfredo sauce; heat through. Drain fettuccine. Stir fettuccine into sauce mixture; heat through. Sprinkle with cheese.

Tips:

Think spring! Primavera is Italian for "spring," and dishes titled "a la primavera" usually contain fresh vegetables. Pasta primavera is the most popular dish in the primavera family.

Too many veggies to chop? Take advantage of the precut vegetables in your supermarkets produce section.

The PCLinuxOS Magazine

Created with Scribus

ms_meme's Nook: I Double Dare You

I double dare you from Windows to flee
I double dare you to try Trinity
Download it now user friendly
Don't be a scaredy cat
What do you care
Take the dare

I double dare you to try it and then
I double dare you to do it again
And if you do there is no turning back
I double dare you
To join the PCLOS pack
I double dare you

MP3

I double dare you from Windows to flee
I double dare you to try Trinity
Download it now user friendly
Don't be a scaredy cat
What do you care
Take the dare

I double dare you to try it like me
I double dare you to do it and see
And if you do there is no turning back
I double dare you
To join the PCLOS pack
I double dare you

OGG

PCLinuxOS Puzzled Partitions

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

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 prefilled 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 252, average score 176.

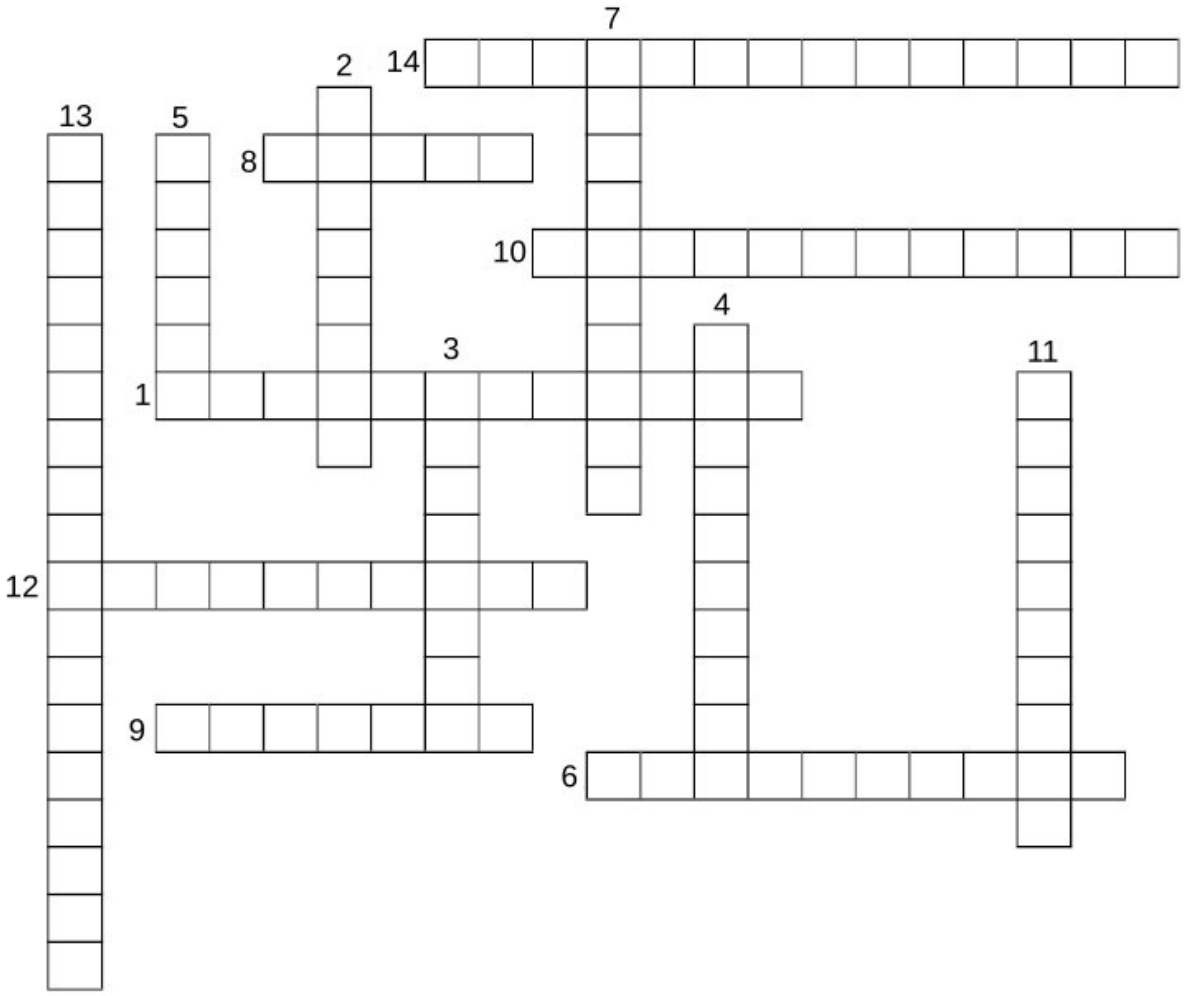
PCLinuxOS Word Find: April 2019
Auto Parts

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

accelerator
alternator
axle
battery
brakes
bumper
camshaft
carburetor
catalytic converter
chassis
clutch
crankshaft
cylinder
dashboard
differential
drive shaft
exhaust system
frame
gauge
hubcap
hybrid
ignition
manifold
muffler
odometer
piston
radiator
shock absorber
sparkplug
speedometer
spoiler
suspension
tachometer
transmission
turbocharger
undercarriage
valve
vents
windshield

Download Puzzle Solutions Here

Auto Parts Crossword



1. a device that splits the engine torque two ways, allowing each output to spin at a different speed.
2. a rod in an engine and works to change circular motion into motion up and down
3. the part of the engine that is filled with water in order to cool the engine.
4. an instrument which measures the working speed of an engine, typically in revolutions per minute.
5. a car that uses more than one means of propulsion; i.e. combining a petrol or diesel engine with an electric motor.
6. the device in an internal combustion engine for mixing air with a fine spray of liquid fuel.
7. used in modern automobiles to charge the battery and to power the electrical system when its engine is running.
8. an instrument or device for measuring the magnitude, amount, or contents of something, typically with a visual display of such information.
9. supplies electrical current to a motor vehicle
10. a mechanism for transmitting power from a vehicle's engine (or motor) to its wheels.
11. includes wheels and tires, springs and shocks, various linkages and joints, and the steering system.
12. transmits the power developed by the engine to the various parts of the vehicle.
13. incorporated in the exhaust system of a motor vehicle, it converts pollutant gases into less harmful ones.
14. the part of an engine that supplies the fuel/air mixture to the cylinders.

Download Puzzle Solutions Here

Mixed Up meme Scrambler

He thought the job was Linux

— — — — — — — — — —

Administrator

Guard
TRENSY

Damp
TIOMS

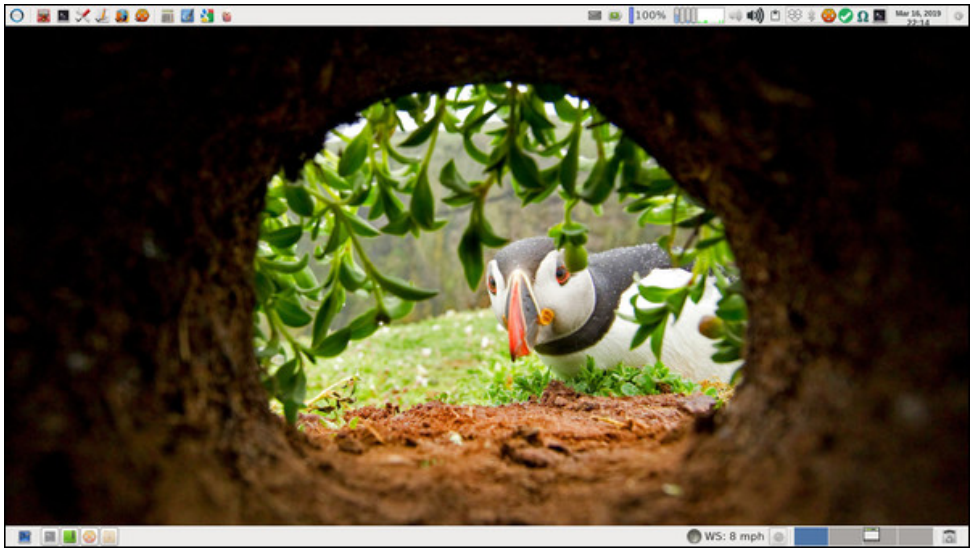
Insect
GANT

Stadium
ARANE

Use the clues to unmix the letters to make a new word. Remix the letters in the red boxes to solve the puzzle.

Download Puzzle Solutions Here

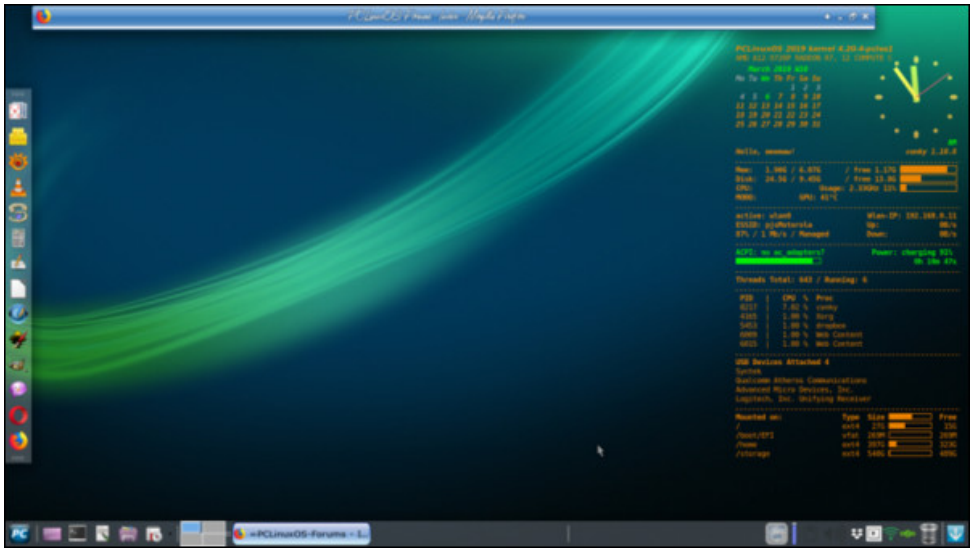
More Screenshot Showcase



Posted by parnote, March 16, 2019, running Xfce.

Posted by mutse, March 3, 2019, running Trinity.

Posted by aguila, March 7, 2019, running KDE.



Posted by Meemaw, March 6, 2019, running Xfce.