

# MUUGLines

The Manitoba UNIX User Group Newsletter

Volume 29 No. 9, May 2017

Editor: Trevor Cordes

## Next Meeting: May 9th, 2017

### Presentation: Intro to Linux on Azure

Microsoft has made Linux a “First Class” citizen on Azure. John Lange will cover some Azure basics, then dive into setting up your first Linux systems on Azure and reveal some of the hidden benefits of using public cloud to run and manage your Linux virtual machines. Time permitting, we will setup a small domain-managed Linux cluster with load balancing and automated management.

### Daemon-Dash: Let’s Encrypt

Adam Thompson will give a short tutorial on Let’s Encrypt: the up-and-coming, and *free* (as in beer), way to obtain hitherto costly SSL certificates for your web servers.

**The latest meeting details are always at:**  
<https://muug.ca/meetings/>

## Where to Find the Meeting

University of Winnipeg, Room 2L17

### NEW ROOM THIS MONTH

Meetings are held in the University of Winnipeg's Lockhart Hall, along Ellice Avenue, West of Balmoral. We can be found this month (and next) in room 2L17. That’s the second floor. Elevators and stairs are available in Lockhart and Manitoba Hall. Doors are usually open by 7:00 pm with the meeting starting at 7:30 pm. Parking is available on the surrounding streets.



See [uwinnipeg.ca/maps](http://uwinnipeg.ca/maps) for further information about parking and access to the campus.

## Dazzling Door Prizes

As usual, we’ll be giving away a few ebooks at the next meeting. We continue with our you-choose-your-title O’Reilly and No Starch ebook giveaways. In addition, this month the choices are expanded to include a new batch of Python-centric books from No Starch Press.

As a reminder, door prize entry is free to every attendee, even non-members. The first draw will select from members only. Subsequent draws will select from all entrants.

## Vinyl Stickers

MUUG continues to give away free vinyl stickers with our shell logo: show up and grab a couple. They look great on laptops, or use them to promote MUUG around your office or school!

## Perl 6 Lives!

This author must apologize: at a 2016 MUUG meeting I called Perl 6 “vaporware”. Reality is, Perl 6, in a stable/production form, was quietly released December 2015; so quietly this Perl fanatic missed it. One could be forgiven, though, since it took over 15 years from the start of design through to a stable release.



Perl 6 is not actually an implementation: it’s a specification based on a test suite. The implementation that has been released that meets the Perl 6 spec (and so is pseudo-de facto Perl 6) is called

Rakudo. Rakudo compiles (not interprets, like Perl 5) and can use MoarVM or JVM for runtime.

Designer Larry Wall has provided a few choice quotes regarding the Perl 6 philosophy:

*In Perl 6, we decided it would be better to fix the language than fix the user.*

*Easy things should stay easy, hard things should get easier, and impossible things should get hard.*

Perl 6 is a massive departure from Perl 5 and should be considered almost an entirely new language. While there is a “compatibility mode” to run old Perl in Perl 6, the Perl 5 interpreter will continue to be included standard in distros for a long time to come, and is probably preferable for old code.

So what’s juicy and new in Perl 6? You get static types, though they’re entirely optional. Subroutines can now have parameter lists, so Perl’s not so weird anymore with its `@_` idiom. Said parameters can be either positional, named (yay!) and “slurpy”; and those can all be mixed as desired. You can pass parameters to arbitrary blocks, and such constructs can be closures.

Sure to confuse us old hands, sigils (like the `$` before scalar variable names) are invariant, so instead of `$array[0]` you now say `@array[0]`. This should make Perl 6 vastly easier for beginners to learn, and also eliminates the crazy `@{foo{bar}}` syntax often required in old Perl. For object-oriented programming, Perl 6 now uses syntax like the rest of the known world, such as the `class` keyword and dot notation instead of old Perl’s arrow. You also get roles, which are interfaces in disguise.

Regex fans won’t be disappointed, though Perl 6 officially dubs them **regexen** and plunks them in a framework called **rules** with the **rule** keyword. The old-style inline regexen that are so convenient are still permitted, showing Perl 6 hasn’t totally lost the plot or ruined the *raison d’etre*.

Other bonuses include: chained comparisons:

```
if 40 < $muugmembers < 75
```

and lazy evaluation:

```
@doesnt_hang_or_crash=0..Inf;  
and junctions (both “and” and “or”):
```

```
if $room eq '1L11'|'2L17'|'1M28'
```

It appears most of the Perl glitterati (Wall, Foy, Conway, etc) are fully onboard with Perl 6 and it is experiencing rapid development and good support. Several new books are forthcoming such as *Learning Perl 6*, and *Migrating to Perl 6*. Want to play with it now? On Fedora: `dnf install rakudo-star`. Lastly, the Perl 6 mascot butterfly “spokesbug”, pictured above, is named Camelia.

## Perldoc Gem of the Month

From `man perlvar`:

```
local $/; # enable “slurp” mode  
local $_=<FH>;# whole file now here  
Remember: the value of $/ is a string,  
not a regex. awk has to be better for  
something. :-)
```

In addition to the humour, the above idiom remains the tersest way to slurp in an entire file in Perl.

## PCRE Takes Over the World

Why settle for substandard, weak and feeble regular expression languages and implementations? That `egrep` syntax is so 1970. Learn and master just one: the one regex to rule them all! I’m talking Perl Compatible Regular Expressions, aka PCRE: Definitely the best thing to come out of the Perl universe.

Starting with version 10.0.5, MariaDB (nee(ish) Mysql) has switched to PCRE. Fancy this?

```
select * from t where a regexp(  
'(?<!badprefix)good(?!badsuffix)')
```

And standard GNU `grep` we all use every day also supports PCRE with `-P`:

```
grep -P 'num(ber){1,3} \d+?'
```

The command `man perlre` is your friend, and so is *Mastering Regular Expressions*, Friedl, O’Reilly Media.

## Wacky Command of the Day

Heard of `tac` (included in Fedora’s `coreutils`)? Hint, spell it backwards. Yes, this command basically does `cat` in reverse. Now, young grasshopper, go and find a real-world use for this. See also: `rev`.

## Surprising Shell Secrets

Did you know that in bash using `[ ]` for conditionals is actually invoking a named executable file: `/usr/bin/[`. Bizarre but true! On Fedora, `[` is owned by the `coreutils` rpm. For giggles, rename that file temporarily and see what breaks!

Whilst `[ ]` and `[[ ]]` seem very similar for use in bash scripts, `[[ ]]` is usually the better choice, as it's a bash built-in, and doesn't require a fork like `[ ]`.

## VLAN on the Cheap?

Need to do some cheap and small VLAN tricks? Have one long CAT6 run you can't easily add another cable to yet want to have two separate subnets to the destination? It turns out you can get a VLAN-capable switch (with monitoring and QoS) for only around \$45! TP-Link's TL-SG105E is its name.

[http://www.tp-link.com/us/products/details/cat-41\\_TL-SG105E.html](http://www.tp-link.com/us/products/details/cat-41_TL-SG105E.html)

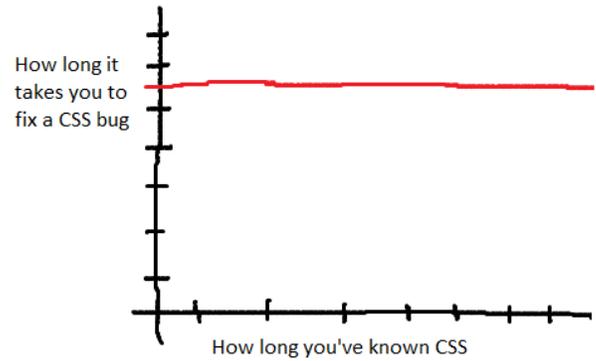
\$45 still too much? How about this crazy hardware hack to turn many old \$10 dumb switches into a VLAN-capable managed switch for just \$2 (and a little soldering). Turns out many dumb switches use the same chips as most routers!

<http://linuxhacking.ovh/2015/08/26/convert-your-unmanaged-to-vlan-capable/>

## CSS Giggles

Some comic relief for those who have ever dabbled in HTML/CSS and website design/programming. From Isaac Lyman at [medium.com](http://medium.com):

*[T]here's no hard-and-fast rule about when to use white-space: nowrap, but multiple studies have found that it's far more likely to work during the fifth hour of debugging than the first. And don't bother trying to outsmart it: CSS can tell when you're just going through the motions (cycling through properties, trying different values for display and position). The clock doesn't start until you're really upset.*



<http://tinyurl.com/koutl9a>

## Amazon Used Book Guide: Update

By Trevor Cordes

In the June 2015 MUUGLines article [How To Buy An O'Reilly Book](#), I discussed how to buy used computer books (especially O'Reilly, No Starch, etc.) on Amazon, often at discounts of 40 to 99%



off cover price. In that timespan I have filled out my personal library quite nicely, at ridiculously low prices, and have picked up a few more tips you may find useful. Firstly, if you are going to (re)read the aforementioned article, please note that **indigo.ca** no longer sells used books.

The **wish list** feature is incredibly helpful for keeping an eye out for deals on all the titles you want to buy. You can use your existing default wish list or create a new one just for the purpose of watching your used-to-buy titles. You can even set your new list to be the default. Then look up each of the titles you are interested in. Make sure you are in the "Paperback" page, and not the Kindle page (click on Paperback if you are not). Click **add to wish list**.

Once you have added a bunch of titles, navigate to your wish list using the nav menu at the top. I find it easiest to click **filter & sort** to sort by title. Don't

Social:



<https://muug.ca/meetings/video>

sort by price, as it only pertains to the new price, not used, and so doesn't help us here. Now you can scan through your list in seconds, focusing on the **Used & New** line of each stanza. When you see one that is \$0-\$15, or similarly cheap in proportion to Amazon's list price, simply click on the **Used** link and see if the cheap copy/copies includes one in a condition that sounds acceptable to you (I usually only shun *acceptable* grade).

In this way you can check your list periodically (i.e. weekly) in about a minute, and pounce on deals that pop up. I have found that pricing can vary dramatically over time on some titles, whilst others remain fairly stable. I have also found that the prices usually reflect changes in the USD/CAD exchange rate, so when CAD is strong you will get better deals. As my original article discussed, shipping is always just a flat \$6.49, and all prices are in CAD, and there are never any hidden "gotcha" fees or S&H-gouging.

Surprisingly, the prices of most titles that I watch increased markedly recently, with no obvious explanation: and not the \$1 or so caused by the exchange rate, but \$10-\$20 on many books. Perhaps "the cat is out of the bag" and more people are discovering the great deals on offer, thus increasing demand. Perhaps Amazon upped their "take". Whatever the cause, hopefully the price surge is transient.

In any event, the above tips can still sniff out deals for you; though increased patience is required. I still regularly find deals at 60-90% off (excluding shipping) and tend to pounce on anything in my wish list above 75% off.

<https://muug.ca/pub/muuglines/pdf/muug1506.pdf>

## Google Guetzli Shrinks JPGs 35%

Google recently open-sourced a new JPEG encoder (for image compression) that can reduce file size by around 35 percent without any perceptible loss in quality! That's 35% compared to the normal JPEG size (i.e. in addition to the compression JPEG usually gives you)!

Even more astounding is this new encoder is compatible with existing decoders, such as those

builtin to existing web browsers. That means that much of the web's imagery just became potentially 35% more efficient without any major pain or upheaval a brand new codec (like WebP) would require.

Run over to Github and check it out. It's probably too early to find a package from your distro vendor, but that situation probably won't last long. The usual caveats apply regarding double-encoding using lossy algorithms but in this case it may be worth converting your old JPGs, especially ones served up on the web.

<http://tinyurl.com/l6l5dqs>

## Join MUUG E-Mailing Lists

Do it now! What are you waiting for! (With apologies to Arnold.) Did you know that MUUG has email mailing lists? Never miss an upcoming meeting again due to lapses in memory (the stuff in your head, not your computer) by joining the muug-announce list:

<https://muug.ca/mailman/listinfo/muug-announce>

That list is ultra-low volume: usually only one email a month. At the very least you can use it to divine which new and secret room you'll be shuffling to that week (we might even give you the password; tell them Guido sent you).

Want help with (or to help with) computer problems, or simply chew the fat? Join the roundtable mailing list. Think of it as an extension of our monthly in-meeting "round table" discussion. It is also fairly low volume, though occasionally gets quite bursty:

<https://muug.ca/mailman/listinfo/roundtable>

There's no downside to joining, and you can easily unsubscribe at any time by revisiting the above links. I recommend **not** selecting the **digest** mode, unless you are thoroughly familiar with (and prefer) it; or you plan on (so-called) lurking.



Except where otherwise noted, all content in this newsletter is licensed under a Creative Commons "Attribution-ShareAlike 2.5 Canada" License.

[http://creativecommons.org/licenses/by-sa/2.5/ca/deed.en\\_CA](http://creativecommons.org/licenses/by-sa/2.5/ca/deed.en_CA)