MUUGLines

The Manitoba UNIX User Group Newsletter

Next Meeting: April 8th, 2014

RTFM and presentation:

For this month's RTFM, Michael Doob will talk about **pdftk** - A handy tool for manipulating PDF.

Want to sell something on the net?

This month Brad Vokey will be presenting on **OpenCart** – a powerful open source shopping cart system that is feature rich and user friendly.

OpenCart is a robust e-commerce solution for internet merchants to create their own online business and participate in e-commerce at minimal cost. It includes an intuitive admin interface that allows you to have complete control over your store. It also has thousands of high end themes available and integrates with hundreds of payment gateways and shipping integrations.

There is also a thriving OpenCart extension market with over 12,000 extensions currently written for it (and more being added all the time). If there is something you need your store to do, there is probably an extension already written for it. If there isn't, or if you don't like the price someone is charging for their extension, you can simply add the code yourself, add it to the OpenCart store, and maybe make a little money selling your extension to others.

MUUG Mugs!

Back by popular demand! MUUG now has new coffee mugs, laser-etched with our age old, lovable logo for sale for $15 cash. There are a limited number available, so get yours at the next meeting!

Where to Find the Meeting

University of Winnipeg Lockhart Hall (marked "L" on the map), on the south-east corner of Spence and Ellice. Parking is available on the surrounding streets. Meetings are normally in room 1L08, but occasionally are relocated to nearby rooms. If there is a change, it should be conveyed via a sign on the door to 1L08.

Computer History Museum Makes MS-DOS and Word for Windows Source Code Available to the Public

On March 25, the Computer History Museum announced that it has, with permission from Microsoft Corporation, made available the original source code for two historic programs: MS-DOS, the 1982 "Disk Operating System" for IBM-compatible personal computers, and Word for Windows, the 1990 Windows-based version of their word processor.

Facebook Open-Sources a New Programming Language

On March 20th, Facebook unveiled Hack, a programming language it developed that integrates seamlessly with PHP.

The social network has migrated its entire PHP code base to Hack over the past year, saying that the new programming language "resolves the fast development cycle of PHP with the discipline provided by static typing, while adding features from other, modern programming languages."

Software Engineer Julien Verlaguet and Security Engineer Alok Menghrajani wrote:

"Every PHP programmer is familiar with day-to-day tasks that can be tricky or cumbersome. The code below is a great example of a common mistake where a method could unexpectedly be called on a null object, causing an error that would not be caught until runtime."

"With thousands of engineers working on millions of lines of code at Facebook, these challenges scaled up. Could detecting errors coexist with rapid iteration, all while preserving our investment in PHP? Could improved code analysis and introspection help make developers more productive with tools like autocomplete?"

"Traditionally, dynamically typed languages provide rapid development while sacrificing the ability to catch errors early and introspect code quickly, particularly on larger code bases. Conversely, statically typed languages provide more of a safety net, but often at the cost of quick iteration. We believed there had to be a sweet spot."

"And, thus, Hack was born. We believe that it offers the best of both PHP and statically typed languages, and that it will be valuable to projects of all sizes."

"We are delighted to open-source both Hack and the tools that we used to convert our code base. We believe this is the beginning of an exciting journey, and we are dedicated to making software development ever easier for both our own engineers and the broader community. Hack's value is not limited to big projects: With type information, good error messages, and fast feedback, small code bases can reap the benefits of Hack, as well."

The verdict from numerous tech sites has been that Hack signals much good news.

Facebook Blog Post: http://bitly.com/muug1404-6

```
                        PHP
 1 <?php
 2
 3 function dbfetch() {
 4   $r = mysql_query('SELECT ...');
 5   if ($r === false || mysql_num_rows($r) == 0) {
 6     return null;
 7   }
 8
 9   return new DBData(mysql_fetch_array($r));
10 }
11
12 function test() {
13   $data = dbfetch();
14   $data->doFunStuff();
15 }
```

```
                        HACK
 1 <?hh
 2
 3 function dbfetch(): ?DBData {
 4   $r = mysql_query('SELECT ...');
 5   if ($r === false || mysql_num_rows($r) == 0) {
 6     return null;
 7   }
 8
 9   return new DBData(mysql_fetch_array($r));
10 }
11
12 function test(): void {
13   $data = dbfetch();
14   $data->doFunStuff();
15 }
```

Open Source version of Hack: http//hacklang.org/

ATM Operators Eye Linux as Alternative to Windows XP

Some financial services companies are looking to migrate their ATM fleets from Windows to Linux in a bid to have better control over hardware and software upgrade cycles.

Windows XP currently powers nearly 95% of ATMs around the world. When Microsoft pulls the plug on support for the operating system on April 8, ATM operators who have not upgraded will essentially be running their systems on an obsolete operating system with no technical support from Microsoft

More than 60% of the more than 400,000 ATMs in the U.S. are expected to be on Windows XP past the April 8 deadline.

Firefox 29 New Look, Improved Browser Synchronization

On March 20th, a day after releasing the stable version of Firefox 28, Mozilla launched the first beta version of Firefox 29. While there wasn't all that much interesting in Firefox 28, the latest beta version represents a major step for Mozilla. It includes both the new - and very Chrome-like - Australis design, as well as support for Firefox Sync, Mozilla's cloud-based syncing service with end-to-end encryption.

The new design will likely be a bit controversial with long-time Firefox users. It's quite a departure from earlier versions, with rounded tabs and a menu that's now behind the same kind of "hamburger menu" as Chrome's. Tech Crunch reporter Frederic Lardinois says, "I've been running the Firefox Nightly releases and Chrome side-by-side lately, and sometimes I get confused about which one I'm using — that's how similar they look."

What Firefox is keeping, though, is the extra search box, which most other browser vendors have done away with. You can always remove it, and just use the URL bar to kick off your searches, too, just like in Chrome or the latest versions of Internet Explorer.

That kind of customization is also at the core of Mozilla's thinking here. The organization thinks of Firefox as a browser you should be able to customize for your needs and to do this, the beta features extensive and easy to find customization options. Earlier versions of Firefox had similar features, but they were often hidden away a bit and most mainstream users never found them. With the Australis design, customization is now front and center.

The design also puts an emphasis on simplicity. Tabs that aren't currently active blend into the background, for example, and unless there is a page to go forward, the design simply hides the usual arrow buttons. Tabs now also sit higher up in the browser, so you get a few extra pixels of screen real estate for the sites you are looking at.

If you want to give the new beta a try, you can download it for Windows, Mac and Linux
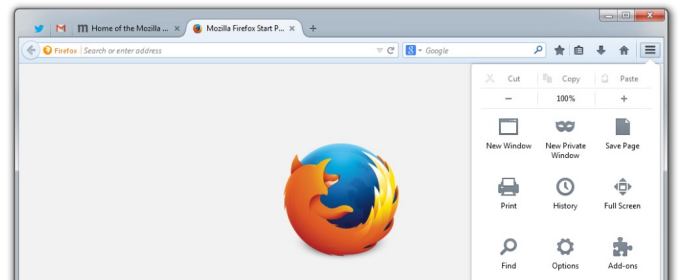
GNOME 3.12: Pixel Perfect... But Homeless

When the GNOME 3.x desktop arrived many said it was, frankly, unusable. Fast-forward three years and while GNOME 3.12, released on March 26, still isn't the infinitely customizable experience of GNOME 2.x, it has progressed by leaps and bounds.

If you dismissed GNOME 3 when it arrived, you may want to take another look – but you will need to be patient. No major distro currently has a release to showcase GNOME 3.12. That's a shame because it may well be the best thing to come out of the GNOME camp in years.

"The universe started in 1970. Anyone claiming to be over 44 is lying about their age."

Perl Finance::QIF to the Rescue!

A doh! moment by: Bradford C. Vokey

In early December, I integrated my Matchmaker program with Canada Post's software to save an additional 2% off our current shipping rates. It was simple enough, and the last step was subtracting the 2% discount from our rates in our Matchmaker program which then, among other things, gets imported into our accounting software.

```perl
price -= price * 0.02;
```

All was good until the end of February when I started reconciling our UPS invoices. None of our UPS shipments since early December were balancing. I had neglected to add a check in the Matchmaker program, to only apply the discount to our Canada Post shipments. Doh!

Now I had 1,580 transactions imported into our accounting software that would not reconcile with our actual UPS invoices. Now what?

Perl Finance::QIF to the rescue!

If anyone ever needs to modify a whole bunch of transactions in pretty much any accounting program, here is how to do it:

1) Export the transactions from your accounting software as a standard .qif file (the Quicken Interchange Format is the de facto export/import file structure of most accounting software packages).

2) Write a quick perl program to fix the transactions.

3) Re-import the transactions back into your accounting software.

Of course you could use any text parsing program (awk/sed anyone?) to do the same job, but why re-invent parsing the .qif format by hand when Finance::QIF does it for you!

Finance::QIF on CPAN: http://bitly.com/muug1404-3

QIF File Format Specs: http://bitly.com/muug1404-4

```perl
#!/usr/bin/perl -w
# QIF File Re-calculator (to fix Brad's doh!)

use strict;
use Finance::QIF;

my $in  = Finance::QIF->new( file => 'in.qif' );          # open input file
my $out = Finance::QIF->new( file => '>out.qif' );         # open output file
my $header = "";                                           # keep track of qif header in case it changes

RECORD: while ( my $record = $in->next() ) {              # parse every transaction in input file
  next RECORD if (                                         # skip transaction if
    exists $record->{status}                               #   already reconciled
       or $record->{payee} =~ /RTN/                        #   a UPS RETURN shipment
       or $record->{payee} =~ /payment/                    #   a UPS payment
       or $record->{payee} =~ /DOMESTIC/                   #   a UPS DOMESTIC shipment
  );
  my $a = $record->{transaction};                          # grab the transaction amount
  $record->{transaction} = sprintf ("%.2f", $a * 1 / .98); # adjust by 2%
  if ( $header ne $record->{header} ) {                    # if header changed
    $out->header( $record->{header} );                     #   output new header
    $header = $record->{header};                           #   keep track of latest header
  }
  $out->write($record);                                    # write new transaction to output file
}
$in->close();                                              # close files
$out->close();
```