Published on *Tux Machines* (http://www.tuxmachines.org)

Programming Leftovers

By *Roy Schestowitz*
Created *17/09/2021 - 9:18pm*
Submitted by Roy Schestowitz on Friday 17th of September 2021 09:18:42 PM Filed under Development [1]

- **Announcement : An AArch64 (Arm64) Darwin port is planned for GCC12** **[2]**

  As many of you know, Apple has now released an AArch64-based version of macOS and desktop/laptop platforms using the ?M1? chip to support it. This is in addition to the existing iOS mobile platforms (but shares some of their constraints).

  There is considerable interest in the user-base for a GCC port (starting with https://gcc.gnu.org/bugzilla/show_bug.cgi?id=96168) - and, of great kudos to the gfortran team, one of the main drivers is folks using Fortran.

  Fortunately, I was able to obtain access to one of the DTKs, courtesy of the OSS folks, and using that managed to draft an initial attempt at the port last year (however, nowhere near ready for presentation in GCC11). Nevertheless (as an aside) despite being a prototype, the port is in use with many via hombrew, macports or self-builds - which has shaken out some of the fixable bugs.

  The work done in the prototype identified three issues that could not be coded around without work on generic parts of the compiler.

  I am very happy to say that two of our colleagues, Andrew Burgess and Maxim Blinov (both from embecosm) have joined me in drafting a postable version of the port and we are seeking sponsorship to finish this in the GCC12 timeframe.

  Maxim has a lightning talk on the GNU tools track at LPC (right after the steering committee session) that will focus on the two generic issues that we?re tackling (1 and 2 below).

  Here is a short summary of the issues and proposed solutions (detailed discussion of any of the parts below would better be in new threads).

- **Apple Silicon / M1 Port Planned For GCC 12 - Phoronix** **[3]**

Developers are hoping for next year's GCC 12 release they will have Apple AArch64 support on Darwin in place for being able to support Apple Silicon -- initially the M1 SoC -- on macOS with GCC.

LLVM/Clang has long been supporting AArch64 on macOS given that Apple leverages LLVM/Clang as part of their official Xcode toolchain as the basis for their compiler across macOS to iOS and other products. While the GNU Compiler Collection (GCC) supports AArch64 and macOS/Darwin, it hasn't supported the two of them together but there is a port in progress to change it.

- **Dirk Eddelbuettel: tidyCpp 0.0.5 on CRAN: More Protect?ion** **[4]**

  Another small release of the tidyCpp package arrived on CRAN overnight. The packages offers a clean C++ layer (as well as one small C++ helper class) on top of the C API for R which aims to make use of this robust (if awkward) C API a little easier and more consistent. See the vignette for motivating examples.

  The Protect class now uses the default methods for copy and move constructors and assignment allowing for wide use of the class. The small NumVec class now uses it for its data member.

- **QML Modules in Qt 6.2** **[5]**

  With Qt 6.2 there is, for the first time, a comprehensive build system API that allows you to specify a QML module as a complete, encapsulated unit. This is a significant improvement, but as the concept of QML modules was rather under-developed in Qt 5, even seasoned QML developers might now ask "What exactly is a QML module". In our previous post we have scratched the surface by introducing the CMake API used to define them. We'll take a closer look in this post.

- **Santiago Zarate: So you want to recover and old git branch because it has been overwritten?** **[6]**

- **Start using YAML now | Opensource.com** **[7]**

  YAML (YAML Ain't Markup Language) is a human-readable data serialization language. Its syntax is simple and human-readable. It does not contain quotation marks, opening and closing tags, or braces. It does not contain anything which might make it harder for humans to parse nesting rules. You can scan your YAML document and immediately know what's going

on.

[...]

At this point, you know enough YAML to get started. You can play around with the online YAML parser to test yourself. If you work with YAML daily, then this handy cheatsheet will be helpful.

- **40 C programming examples [8]**

  C programming language is one of the popular programming languages for novice programmers. It is a structured programming language that was mainly developed for UNIX operating system. It supports different types of operating systems, and it is very easy to learn. 40 useful C programming examples have been shown in this tutorial for the users who want to learn C programming from the beginning.

Development

Source URL: http://www.tuxmachines.org/node/155767

Links:
[1] http://www.tuxmachines.org/taxonomy/term/145
[2] https://gcc.gnu.org/pipermail/gcc/2021-September/237340.html
[3] https://www.phoronix.com/scan.php?page=news_item&px=GCC-12-Apple-M1-Port-Plan
[4] http://dirk.eddelbuettel.com/blog/2021/09/17#tidycpp_0.0.5
[5] https://www.qt.io/blog/qml-modules-in-qt-6.2
[6] https://foursixnine.io/blog/linux/tech/git/2021/09/17/So-you-want-to-recover-and-old-git-branch-because-it-has-been-overwritten.html
[7] https://opensource.com/article/21/9/intro-yaml
[8] https://linuxhint.com/40-c-programming-examples/