Programming Leftovers

By *Roy Schestowitz*
Created *21/01/2021 - 9:49pm*
Submitted by Roy Schestowitz on Thursday 21st of January 2021 09:49:11 PM Filed under Development [1]

- **What is your favorite Linux scripting or programming language? | Enable Sysadmin [2]**

  Of all the scripting and programming language options available to you on the Linux platform, which one is your favorite?

- **When costs are nonlinear, keep it small. [3]**

  It shows Preventive Maintenance as a series of small costs. Minor repairs are a series of bigger costs, and major repairs are much bigger than that. Every maintenance delayed escalates into a minor repair and then a major repair ? costs increase nonlinearly with time delay.

- **Old compilers and old bugs [4]**

  The kernel project goes out of its way to facilitate building with older toolchains. Building a kernel on a new system can be enough of a challenge as it is; being forced to install a custom toolchain first would not improve the situation. So the kernel developers try to keep it possible to build the kernel with the toolchains shipped by most distributors. There are costs to this policy though, including an inability to use newer compiler features. But, as was seen in a recent episode, building with old compilers can subject developers to old compiler bugs too.

  On January 5, Russell King reported on a problem he had been chasing for a long time. Some of his 64-bit Arm systems running 5.4 or later kernels would, on rare occasion, report a checksum failure on the ext4 root filesystem. It could take up to three months of uptime for the problem to manifest itself, making it, as King described it, "unrealistic to bisect". He had, however, found a way to more reliably reproduce the failure, making the task of finding out

when the problem was introduced plausible, at least.

Starting with King's findings, a number of developers working in the Arm subsystem looked into the issue; their efforts seemed to point out this commit as the culprit. That change, applied in 2019, relaxed the memory barriers used around I/O accessors, optimizing accesses to I/O memory. Reverting this patch made the problem go away.

- **Callback Function in C++ ? Linux Hint [5]**

  A callback function is a function, which is an argument, not a parameter, in another function. The other function can be called the principal function. So two functions are involved: the principal function and the callback function itself. In the parameter list of the principal function, the declaration of the callback function without its definition is present, just as object declarations without assignment are present. The principal function is called with arguments (in main()). One of the arguments in the principal function call is the effective definition of the callback function. In C++, this argument is a reference to the definition of the callback function; it is not the actual definition. The callback function itself is actually called within the definition of the principal function.

  The basic callback function in C++ does not guarantee asynchronous behavior in a program. Asynchronous behavior is the real benefit of the callback function scheme. In the asynchronous callback function scheme, the result of the principal function should be obtained for the program before the result of the callback function is obtained. It is possible to do this in C++; however, C++ has a library called future to guarantee the behavior of the asynchronous callback function scheme.

- **Regarding the closure of rt.cpan. [6]**

  Let me preface this short post with this, I don't have the solution to this problem. Perhaps there is someone in the wider Perl space who is well placed to pick this up, but there seems to be little going on in terms of community engagement.

  In the first week of 2021 I noticed a link to this sunset message for rt.cpan behind displayed on the rt.cpan homepage. Firstly I believe the notification on the page could be highlighted better, grey on grey, on a page with lots of grey isn't exactly eye catching.

  At the time the linked article didn't contain much information, besides a date. It has since been updated with links to resources to migrate tickets elsewhere.

  A reply to my post in the perlmonks news section was concerning to me, I shortly found the infrastructure working group post on topicbox (which I find no link to on any of the perl websites, or release documentation). This thread was concerning in so much as a single

volunteer has decided to step back, which is of course fine, but it doesn't seem like the option of asking the wider community if anyone would be willing to step up and take it over has been explored. It doesn't even seem to be being openly discussed.

- **Perl weekly challenge 096 - Raku** **[7]**

- **GNU Linux Bash ? script for troubleshoot long term test testing network internet connection connectivity** **[8]**

  this script is intended for long term testing of reliability of network connection/connectivity

Development

Source URL: http://www.tuxmachines.org/node/146727

Links:
[1] http://www.tuxmachines.org/taxonomy/term/145
[2] https://www.redhat.com/sysadmin/favorite-linux-language
[3] https://jessitron.com/2021/01/18/when-costs-are-nonlinear-keep-it-small/
[4] https://lwn.net/Articles/842122/
[5] https://linuxhint.com/callback-function-in-c/
[6] http://blogs.perl.org/users/martin_mcgrath/2021/01/regarding-the-closure-of-rtcpan.html
[7] http://blogs.perl.org/users/joan_mimosinnet/2021/01/perl-weekly-challenge-096---raku.html
[8] https://dwaves.de/2021/01/21/gnu-linux-bash-script-for-troubleshoot-long-term-test-testing-network-internet-connection-connectivity/