CAPLin framework to build a SocketCAN node application in C

The SocketCAN functionality, combined with the can-utils programs, enable you to view, interact and analyze the CAN bus traffic on Linux. However, these tools are no match for high-end tools such as Vector CANalyzer and CANoe under Windows. I especially miss CAPL scripts on Linux. For this reason I developed the CAPLin framework. With CAPLin you can quickly build a SocketCAN node application in the C programming language.

Thoughts on how to find remote work in Cameroon

Remote work is the new norm there has never been a time like this, where as a SE you can make more than a decent living. This isn?t a know-it-all kind of post, I just wanted to write a bit about my experience, but it?s way too long (6 months+) so I will just share what worked and not for me. Before I forget, this is mainly for people like me doing computer science for the sake of doing it. Not because someone forced us or whatever. In short, geeks I guess. If you?re like me the perspective of spam applying and writing corresponding CVs is not very appealing. So, if CS is just a means to an end - not that there?s something wrong with that - but this might rub you off the wrong way (and you guessed right, no, I don?t look forward to enter management to ?escape? coding). The job landscape in Cameroon is? saddening. While everywhere else the supply exceeds the demand, here it?s the exact opposite, which inevitably leads to abuse. Also, if you are still a student, this might not be for you directly, you can still read it to be prepared but there are many opportunities for students and I talk a bit about GSoC here. That being said, let?s get started.

Emacs is a Lifestyle
I think that perfectly captures the spirit of Emacs and the nature of its (most devoted) users. I'd even go a bit farther and make the claim that (using) Emacs is essentially a lifestyle (choice).

- **Uninitialized Stack Variables** [5]

  Finally, as we observe here once more, writing C leaves us (necessarily) at the whims of the compiler: FreeBSD 13.0-RELEASE happens to use clang, and gcc(1) would have failed in either of our two scenarios. So one question that arises is whether compilers should perhaps auto-initialize stack variables.

  clang has a discussion around this, as does gcc, but there does not seem to be an agreed upon conclusion. Considering the possible security implications, it does seem to me that it would be a Good Thing? to at least move away from having uninitialized variables by default and instead requiring explicit requests from the programmer (say, by way of an attribute?) that a given stack variable not be initialized. But I honestly don't know what the performance impact of this would be.

  Either way, I'm going to make it a habit to memset(3) my structs going forward...

- **Testing** [6]

  I think about tests in terms of defense in depth, value-for-effort and debugging efficiency.

  Debugging efficiency is not something I see discussed often and it's the only place where I disagree slightly with Aleksey's post above. The more that happens between the cause of a bug and the actual test failure, the longer it takes to track down the bug. So I tend to write unit tests for code which is: [...] 

- **Writing** [7]

  I have a file called 'ideas' where I write down potential projects or thoughts that might be worth writing about. Entries grow over time as I add more thoughts. The entry that eventually became Against SQL existed for over a year. Every time I encountered a new bizarre corner of the SQL I would make a quick note of it.

  Eventually one of the ideas will feel ready and I'll try to write it up in full. This can take anywhere from a few hours to a few weeks depending on what the goal of writing it is and how much research is required. Against SQL took something like 60-80 hours to write because I was trying to make a strong argument about a complicated and contentious subject. Why
isn't differential dataflow more popular took maybe an hour or two because I just wanted to hear about other peoples experiences.

- **Property-Based Testing In Go [8]**

  Property-based testing can be a bit trickier to learn, and not every problem can be well tested in this manner, but it?s a powerful technique that?s well supported by the go std-lib (testing/quick) and that is under-utilized.

- **[Old] EP. #91: Open Source Security: with Dr. David A. Wheeler [9]**

  In episode 91 of The Secure Developer, Guy Podjarny speaks to Dr. David A. Wheeler, an expert in both open source and developing secure software. David is the Director of Open Source Supply Chain Security at the Linux Foundation and teaches a graduate course in developing secure software at George Mason University. Today?s discussion revolves around open source security (or OSS), in which David is an expert, not just from the perspective of consuming open source but also creating and even governing open source. Tuning in, you?ll learn about some of the primary security concerns in open source and the necessity to educate developers about secure software.

- **[Old] Managing Risks and Opportunities in Open Source with Frank Nagle & David A. Wheeler [10]**

  We start off on the topic of looking at metrics that are useful for identifying what?s going on in a Software Configuration Management system. David tells us what it is and if there?s a difference between building software and deploying it. Also, figuring out which components you?re going to bring in, to your overall system.

- **Toit open-source language claims to be 30x faster than MicroPython on ESP32 - CNX Software [11]**

  Developed by a team of former Google employees, Toit is a complete IoT platform with remote management, firmware updates for fleets of devices with features similar to the one offered by solutions such as balena, Microsoft Azure, or Particle edge-to-cloud platform.

  Toit currently works on ESP32 microcontrollers using lightweight containers, and after seeing existing high-level languages MicroPython and Javascript were not fast enough on low-end microcontrollers platforms, the team at Toit started to develop the Toit language in 2018, and
has just made it open-source with the release of the compiler, virtual machine, and standard libraries on Github under an LGPL-2.1 license.

**XOR Two Strings in Python** [12]

You may have used many logical, arithmetic, and comparison operators within mathematics and programming while working. One of the frequently used logical operators is the XOR operator. It returns exactly the opposite of the result of the OR operator. Within this article, we will be using the XOR operator on two string-type variable values while working in a Python environment. Let's have some examples in the Ubuntu 20.04 system.

**Python String to a Dict** [13]

In Python, the conversion of different data types is a common problem and it is very important to do it right. Dictionary is the data type that saves the information/elements in a pair form. It is important to convert the string data type to a dictionary data type during programming. However, before going to the methods of conversion, let me explain the strings and dictionaries.

A string is a series of elements in Python. It is unchangeable. The elements or items are enclosed in single and double quotation marks. Since Python has no proper character data type. However, any character is also taken as a string in Python.

In Python, a dictionary is essentially a collection of changeable data items. This collection is present in an unordered form. Dictionaries save the data in which every element is in the form of a pair. The elements inside the brackets are present in the form of pairs and each pair is segregated by the comma. But the elements are isolated by using a colon.

The main attribute of the dictionary is that it does not accept polymorphism. We can get the data from the dictionary later by referencing the appropriate key name. Let's discuss the techniques of converting the string to a dictionary.

**Python String Decode Method** [14]

The Python language is used to store the string in the form of Unicode. Within Unicode, a simple code point is utilized to represent a single character of a Unicode. We have to know two terms: encode and decode. The encoding would convert a simple string to a group of bytes while decoding will convert the group of bytes to a real string once again.

So, within this article today, we will be decoding a string to an original one with the encode()
and decode() function. Be sure to configure the python3 package on your Linux system. Let’s start today’s article by launching the terminal console using the Ctrl+Alt+T.

**Python Removes Newline From a String** [15]

In Python, the strings are a series of elements. These elements are surrounded by single and double quotation marks. Python has a newline symbol. It is represented by `\n`. It is utilized to track the climax of a line and the appearance of a new line. The newline character is utilized in f-strings. In addition, the print statement prints a newline character to the end.

Newline character `\n` is a special character. It is helpful to make a new line. When we utilize the newline character `/n`, a new line is created spontaneously.

**Laravel 8.73 Released | Laravel News** [16]

The Laravel team released 8.73 with support for Countable objects in the string pluralizer, allowing closures for determining cache TTL, a lazyByIdDesc() query builder method, and the latest changes in the v8.x branch.

**Medical Web Development: Top 10 Programming Languages Used in Health Tech** [17]

**What are Container Classes C++?** [18]

A container class as the name suggests is used to contain different values, objects, and variables, etc. in the memory or the external storage. A container class supports other classes present in the programs and it functions to hide the objects/variables used in the memory. It stores many items and all of these items are easily accessible by other members of the program.

All container classes access the elements of the container efficiently through the iterators. This class is known to hold some similar and mixed objects in the memory. A container can be of a homogeneous or heterogeneous type. If the container holds mixed objects then it is heterogeneous, while in the case of similar items it is known as homogeneous container class.

We are going to explain this concept on the Linux operating system, so you need to have Ubuntu installed and in the running form on your system. So you must install Virtual Box and after downloading and installation now configure it. Now add the Ubuntu file to it. You can access Ubuntu’s official website, and download the file according to your system requirement and operating system. It will take hours, then after installation, configure it on the virtual
machine. In the configuration process, make sure you have created the user because it is essential for any operation on the Ubuntu terminal. Moreover, Ubuntu needs the authentication of the user before doing any installation.

We have used the 20.04 version of Ubuntu; you may use the latest one. For the implementation, you need to have a text editor and must have access to the Linux terminal, because we will be able to see the output of the source codes on the terminal through the query. The user must have basic knowledge of C++ and object-oriented programming to make use of classes in the program.

* How to Convert Java to Kotlin and Kotlin to Java [19]

This article will cover a guide on converting code written in the Kotlin programming language to Java programming language and vice versa. Kotlin is a relatively new programming language being developed by JetBrains and it is fully interoperable with Java programming language. It offers some benefits over Java programming language like a more concise syntax, more built-in helper functions, stricter null type checking, data classes, and so on. Full list of differences between these two languages is available here. Kotlin is now the preferred language for developing Android apps and it has been fully integrated into Android Studio app development software suite.

You can convert Kotlin to Java and Java to Kotlin using offline tools. Some of them are explained in this article. Do note that depending on the code being converted and the type of tool being used for the conversion purpose, the converted code may not be 100% accurate and you may have to make some manual edits. You should always review converted code before using it in an application.

**Development**

**Source URL:** [http://www.tuxmachines.org/node/158470](http://www.tuxmachines.org/node/158470)

**Links:**
[3] [https://rmpr.xyz/remote-work-cameroon/](https://rmpr.xyz/remote-work-cameroon/)
[5] [https://www.netmeister.org/blog/stack-vars.html](https://www.netmeister.org/blog/stack-vars.html)
[6] [https://www.scattered-thoughts.net/writing/testing/](https://www.scattered-thoughts.net/writing/testing/)
[7] [https://www.scattered-thoughts.net/writing/writing/](https://www.scattered-thoughts.net/writing/writing/)
[8] [https://earthly.dev/blog/property-based-testing/](https://earthly.dev/blog/property-based-testing/)
[10] [https://podcast.chaoss.community/10](https://podcast.chaoss.community/10)
[16] https://laravel-news.com/laravel-8-73-0
[18] https://linuxhint.com/container-classes-cpp/