

Re: Programmer's unpaid overtime.

Source: <http://coding.derkeiler.com/Archive/General/comp.programming/2003-12/0776.html>

From: Programmer Dude (*Chris_at_Sonnack.com*)

Date: 12/09/03

Date: Tue, 09 Dec 2003 16:36:04 -0600

"Edward G. Nilges" wrote:

- > [...*"reasoning" snipped...*]
- >
- > *Chris Sonnack's numbers show what may be but probably isn't a*
- > *problem in this reasoning,*

The problem is not the reasoning—which, since no hash tables were used in the software under discussion, is moot anyway—but with your (error-ridden) design and (slow) implementation.

The reason for both errors appears to be—as far as I can tell—ignorance of design issues and of your tools. The simple fact of the matter is—with very little effort—I wrote counter-example that blew the doors off your precious PowerString POC.

- > *...my main concern in the real world is not some absolute*
- > *efficiency of a specific program, but the efficiency "trace",*
- > *the efficiency "footprint" of a family of related solutions,...*

No comment. It was just so ... unique ... I had to quote it.
(It's such a great example of your ability to BS.)

- > *I was not completely happy with the usual solution to this problem*
- > *which is to transform all the words once and for all into an array,*
- > *because in real applications the words "on the left" are usually*
- > *more important. A financial news text coming over a wire and being*
- > *parsed for words of interest to the user such as "money", "stocks",*
- > *"equity", and "prison time", might not always be completely read,*
- > *and the appearance of words to the left is more important.*

I would disagree with this reasoning, because you cannot insure that words of interest are not "on the right" until you read them. Thus, in most real world applications, you do end up parsing all the words.

(In fact, it is highly unusual—in my Real World experience—to ever tokenize *part* of a string, but I cannot claim to having worked in

all possible domains).

> [...S*N*I*P...]

>

> *Chris Sonnack presented some interesting numbers. But I have no way of auditing those numbers. He may have made some stupid error.*

All source code was presented along with those numbers. If you failed to investigate the situation, you have only yourself to blame. Source code is still available upon request.

> *I was trying in fact to show how bad C sucks. This is because it completely obscures the difference between code with state and code without state.*

No, this is more of your BS. ALL code has state of some sort, so there is no such thing as "code without state". You seem to be complaining that C lacks classes (or their equal) as a mechanism of encapsulating "state". That may be so, but C doesn't claim to provide such a mechanism, nor does such a mechanism guarantee safety from bad code.

> *If you create an array or cache in the C parser, you have to figure out where to put it. If you put it in "open code" outside of any function [...] you have a New Thing which needs RAM and which within this RAM develops a state.*

(Word: "open code" == "file scope" If you use the proper terms, you won't have to explain what you mean to everyone.)

This is the same in any language any time you need to allocate an object. You seem to be complaining that C requires you to know a little more about what you're doing.

Consider a language like Java or VB with garbage collection. What is the state of the object I just finished using? Is it still around in some fashion? Who knows? Answer: NO ONE!

> *This Thing becomes responsible for managing different strings from different callers. Over and above the individual cache for a string it is responsible for storing and looking up different input strings.*

>

> *Whereas there's a one-to-one relationship, tightly coupled, between a PowerString instance and its string.*

One can easily implement a solution in C that also couples the data with its instance. In fact, that would often be preferred.

The problem, Ed, is that you are criticizing something you do not truly understand or have any real skill with. The only thing your

comp.programming: Re: Programmer's unpaid overtime.

efforts have *clearly* demonstrated is that *you* shouldn't use C.

```
--  
|_ CJSonnack <Chris@Sonnack.com> _____ | How's my programming? |  
|_ http://www.Sonnack.com/ _____ | Call: 1-800-DEV-NULL |  
|_____ |
```