

Re: A note on computing thugs and coding bums

# Re: A note on computing thugs and coding bums

---

*Source:* <http://coding.derkeiler.com/Archive/General/comp.programming/2008-01/msg00516.html>

---

- *From:* spinoza1111 <spinoza1111@xxxxxxxx>
  - *Date:* Sat, 12 Jan 2008 07:17:10 -0800 (PST)
- 

On Jan 12, 6:12 pm, Richard Heathfield <r...@xxxxxxxxxxxxxxxx> wrote:

spinoza1111 said:

On Jan 12, 3:48 pm, Richard Heathfield <r...@xxxxxxxxxxxxxxxx> wrote:

[In deference to Philip, all Nilges's polemic, rhetoric, abuse, etc has  
[been  
snipped after only the briefest scanning to identify its nature, without  
actually reading it; this reply focuses purely on the technical part of  
Nilges's reply, which is of course broken and needs addressing. CAVEAT:  
I have not bothered to indicate where polemic, rhetoric, abuse and other  
such idiocies have been snipped. Refer to the parent article if you care  
enough.]

If you don't read the posts, you have no right to reply.

I have only replied to the parts of the post that I read. I didn't bother to reply to the parts of the post that I have not read. If you had bothered to read what I had written (above), you would have realised this. Since you did not realise it, either you had not read what I had written (in which case, by your own argument, you had no right to reply to it), or you did not understand it, in which case the proper course was to ask for an explanation of the part you did not understand.

If you don't read all the posts, you have no right of reply.

Re: A note on computing thugs and coding bums

In the rest of this reply, I focus purely on the technical aspects, as before. All your polemic, rhetoric, abuse, and other such idiocies have been snipped, unmarked. If you want to read them again, by all means refer to the parent article to this reply.

spinoza1111 said:

On Jan 12, 2:50 pm, Richard Heathfield  
<r...@xxxxxxxxxxxxxxxxxx> wrote:

But I think you [i.e. Ben Bacarisse] need a rider on your claim. I'm sure you meant that the code is valid for any character set that is legal in C (which is a very unrestrictive language, but which does impose /some/ restrictions on character sets – specifically, that the values of characters in the required source character set (other than null) must be positive, that null must be 0, and that '0' to '9' are contiguous).

Non restricted ASCII characters beyond 127 aren't positive if bytes are signed.

There aren't any ASCII characters beyond 127. It's a 7 bit code. We covered that, remember? But to answer your substantive point, yes, if bytes are signed AND are only eight bits wide, then this does

Re: A note on computing thugs and coding bums

mean that  
all the required source character set characters must have  
values in the  
range 0–127 (with the 0 obviously being assigned to the null  
character)..

This not only is a [...] slap in the face to international users,  
it's a [...] slap in the face to American users and an arrogant  
piece of [...].

(Expletives deleted.) C does not require that implementations use ASCII. It allows them to use ASCII. If an implementor wants to use some other character set (EBCDIC, Unicode, some other standardised set, or even a custom character set), the C Standard endorses that decision subject only to the very minor restrictions that: (a) the null character is the character whose bits are all set to 0; (b) the required character set characters must have positive values; (c) the digit characters '0' to '9' have values that are contiguous and which ascend in the obvious way, with '0' first and '9' last within the digit group. This is not a slap in anyone's face, but a broad and liberal approach to character sets.

This is nonsense. C can pretend to be laissez-faire about new code, although anyone who starts a real project in C is an incompetent.

But a "standard" "permits" alternative encodings NOT when it is laissez-faire and says that new code can do all sorts of shit.

A standard permits alternative encodings when an existing program, written in the standardized language following the standard, has a snowball's chance in hell of working for a new encoding with a minimum of fuss approaching zero!

You don't understand encapsulation in the slightest.

A String, in C Sharp or Java, can be redefined. Existing standard-conformant code will work after due diligence. Period.

By due diligence, I DON'T mean calling some Fat Bastard in as an overpriced consultant.

I mean that a regex written for grown-up Strings in C Sharp will increment an index by ONE and index to specific chars which can also be redefined with a minimal amount of due diligence, and no visits by Fat Bastards, pompous garden gnomes, half-crazed monks with silly beards and irritating baritones, or other "C consultants".

## Re: A note on computing thugs and coding bums

It would NOT be if you would only DESIST from claiming that C handles strings.

IT DOES NOT.

C handles strings according to its own definition thereof. If you choose to use another definition of the word "string", that's entirely your choice, but C's definition of "string" is the definition I use when discussing strings in a C context. If you want to discuss your belief that C fails to allow programmers to handle some other data format, I suggest that you define the data format in question in a way that makes it clear what you are talking about.

You have no conception of what a string is, because in 1971, nobody knew what a string was. IBM thought it was an 8, later 16 bit length followed by a string which could include Nul (not "null"). Probably just to pimp IBM's, and perhaps Multic's, ass, Kernighan and his merry men decided to go for "unlimited length strings" while making an exactly equivalent mistake from the standpoint of usability: the fact that you can't process real strings on behalf of humanity that contain...Nul.

Strangely this problem had been solved, and the solution, and the people who developed it, thrown away. The IBM 1401 business midrange computer, announced in 1959, simply added an extra bit to each (6-bit) byte, creating a 7 bit addressable word such that a "word mark" indicated string end. I floored my math professor by using this simple feature in 1973 to compute, possibly for the first time in the history of the human race, probably for the first time in Chicago-land, the exact value of 100 factorial.

[Nominations for my Nobel may now be submitted har har.]

But, for the thoughtcrime of insisting that this was kewl and should be built upon and not scrapped, John Haanstra was subject to IBM Gulag for many years, and IBM's first PL/I compiler required strings to be limited to 255 bytes.

What is a string? It is a sequence of characters or ideograms for serving human needs.

Re: A note on computing thugs and coding bums

You forgot to add that Nul (not "null",  
please, that names something  
different)

In C, the Standard defines the null character as the character  
with  
value 0. More formally: "A byte with all bits set to 0, called  
the null  
character, shall exist in the basic execution character set" –  
C89:  
2.2.1

The C standard is poorly worded. No wonder you like it.

The ISO C Committee does not claim to be perfect, and so there is a process  
for raising faults. You are free to raise a fault (a "Defect Report", or  
DR) if you wish. If the Committee considers that your report has merit,  
they may well decide to change the wording, and no doubt they will take  
into consideration any proposed wording change that you might care to  
suggest.

Thank you for a courteous tone here, but I think I'll take a pass on  
that. I don't think C should be standardized, I think it should be  
shit canned, and you don't want to be responsible for unleashing me on  
the committee.

I have hopes this evening based only on your not vandalizing my latest  
thread on the project status that you have decided to behave yourself  
in future, as well as, just possibly, Randy Howard. I will not mention  
you in new threads or posts to encourage this healthy development.

Since the context is C, I am using C terminology. If you wish  
to use the  
word "null" for something else instead, that's up to you, but I  
have  
made my usage precisely clear by citing the source of my  
definition. I  
have no qualms about using the appropriate term ("the null  
character",  
sometimes abbreviating it to "null" just as we abbreviate "the  
space  
character" to "space", etc), whether you like it or not.

## Re: A note on computing thugs and coding bums

In usable standards, it is Nul to disambiguate it from a null integer returned to indicate failure, which may or may not be zero.

The C89 Standard is considered usable by a great many implementors (since they *\*have\** used it to produce conforming implementations) and a great many programmers (because they *\*have\** used it to produce very portable programs). That you do not consider it usable is perhaps relevant to you, but your opinion regarding its usability is not the only one.

C isn't portable. It wasn't designed to be portable. It was designed by people who didn't grasp the higher power of mathematics to address time and space efficiency to get around this failure by exploiting specific details of machine architecture.

Specific C programs can be very portable. Linux is one example. They use highly specific techniques including a virtual or shadow form of object-orientation enforced at the level of file structure, coupled with primitive curses in Finno-Ugric on people who do stupid and nonportable things for no reason, and macros, to get to this portability.

Demotic C code, on the other hand, has at a minimum to be audited by a Fat Bastard of a C consultant line by line in a way C Sharp does not. This is why I think you promote C, although you may or may not be a Fat Bastard, or dead sexy for that matter.

In C, an integer return value of 0 typically indicates success. For

Yes. I know. This was Yet Another world-historical brain fart from the C team.

example, if we check the `<stdio.h>` header, we find that the following functions use 0 to indicate success: `remove`, `rename`, `fclose`, `fflush`, `setvbuf`, `fgetpos`, `fseek`, `fsetpos`. Functions that do not return 0 for success are either void functions, or functions that return a count of some kind (e.g. the number of characters printed by `printf`), or functions that return a character value (e.g. `putc`, `getc`, `ungetc`), or functions that return a pointer value. (I think that list is exhaustive, but I won't swear to it.)

## Re: A note on computing thugs and coding bums

This is barbarism. The only value that should be returned for success is bool true. Period. The only value that should be returned for failure is bool false. Full stop.

must occur somewhere oh what the hey after  
the character  
set or some stupid thing will happen.

No, it must have the value 0, and must be a part of the character set.  
Because some code point values must follow it (i.e. have positive values), it cannot be at the end, so in no sense does it come "after the character set".

Your failure of comprehension caused you to miss the typo. Substitute string for set and stop wasting my time.

Character sets were the subject of discussion. If you randomly introduce a discussion of character strings into a discussion about character sets and then describe the character strings as character sets, it is you who waste my time, rather than vice versa.

I've already said I was talking about strings but under the stress created by your thread vandalism, which I hope has ceased, I typed set. The confusion results from the fact that I disambiguate sets and strings in an object oriented fashion in my new code, and type the two words frequently.

The problem is that real STRINGS cannot contain NUL. I am well aware that the C character SET contains NUL.

I really hope you know the difference between a character set and a character string. Using the standards of uncharity in force in this community, I would have gotten mediaeval on your ass for what appears to be a simple mistake in globally grasping my meaning and being able to spot my typo.

D'oh (Homer's null utterance).

## Re: A note on computing thugs and coding bums

It is unprofessional to just  
assume that the character string will be terminated by a Nul.

In C, if a sequence of characters is not terminated by a null character, it  
is not a string.

This is absurd! Hamlet is terminated by "Let the soldiers shoot",  
followed by a period. My book is terminated by "The rest is  
television", followed by a period. War and Peace is terminated by "In  
the first case it was necessary to renounce the consciousness of an  
unreal immobility in space and to recognize a motion we did not feel;  
in the present case it is similarly necessary to renounce a freedom  
that does not exist, and to recognize a dependence of  
which we are not conscious."

C does not define reality and is completely inadequate for new  
development!

Death to C!

You skipped most of the post,

That's because most of the post was rhetoric, polemic, abuse, or other such  
idiocy. I made it clear at the outset that I would not address such things  
in my reply.

This makes you ne kulturney. Where do you get off? It was precisely my  
fear as an adolescent, and what prompted me to take my first computer  
science class, that tech thugs would impose their thug vision on  
humanity. C brought this to pass.

and you are such a poor reader that you  
did not mentally correct me when I [...] typed set when I meant string.

So when you make mistakes, it's suddenly my fault? I don't think so.

## Re: A note on computing thugs and coding bums

Here, it was. You created data smog. When you desist, productive discussion will move forward.

How DARE you even reply?

We all have the right of reply. Your pathetic threat of physical violence

You're lying. No such threat was made. I suggested a dialog in a pub. If you like to come to blows, that's your problem. Try me.

has failed to intimidate me. I dare to continue to correct your errors because you dare to continue to make errors. If you don't want me to post corrections to your articles, don't post errors in those articles.

I'd suggest instead that you stay out of any threads in which I am a participant. That would allow you to continue your use of usenet for commercial gain.

You need only not mention me, and I, you.

I've already resorted to referring to you and Randy as the Beastie Boys. Would you prefer Fat Bastards? I look forward to posting here without even that need.

Seriously: don't talk about me and don't talk to me. Don't hire me and I won't hire you.

Or, continue the overall tone of this post in which you make, for the most part, and given your limitations, a genuine effort to explain your world-view.

Sure, I use profanity and hyperbole. You don't. And, you do a reasonably good job in explaining modern C.

But, you don't understand its context. It's a flawed programming language that never (in the manner of many of its Baby Boom programmer generation) grew up.

The Algol team realized through actual experience in a Europe still recovering from WWII that computing was going to be hard. The United States, and IBM, didn't want to hear this because it was entering a silly season in which we, the boomer generation, would be able to grow stupid beards and not wear socks to work, and stay close to machines, which were easier to deal with than the men of my father's generation,

Re: A note on computing thugs and coding bums

Re: A note on computing thugs and coding bums

damaged goods as they were by the experience of the war.

I'm serious. "Staying close to the machine" and being able to twiddle bits was and is a Peter Pan gesture, and a way to not grow up.

The Algol team, grown-ups like Dijkstra, realized that if they were not given enough time by uncaring American corporations, which at the time had the European economy by the balls, society would eventually collapse under the weight of unmastered data systems, which it is doing post-September 11 (where the FBI team investigating al-Qaeda wasn't given a query engine capable of Boolean logic) and in the current subprime mortgage crisis (where it appears to me that programmers could not be bothered to, or were scared to, search chains of virtual or real derivatives for credit or reinsurance spirals).

The PROBLEM is that today, if you talk like a 19th century adult male, you DO sound like a raving maniac, because it's accepted as a social axiom that society is beyond our control. This is a legacy of the collapse of Marxism.

--

Richard Heathfield <<http://www.cpax.org.uk>>

Email: [richard@cpax.org.uk](mailto:richard@cpax.org.uk)

Google users: <<http://www.cpax.org.uk/prg/writings/googly.php>>

"Usenet is a strange place" – dmr 29 July 1999– Hide quoted text –

– Show quoted text –