

Why C Is Here to Stay

Steve Oualline

Dec 18, 2001

With all the new languages coming out, is the C language, which is over thirty years old, still relevant? The answer is still a resounding “Yes!”.

Let’s look at the software we have today. I’m writing this column using the Vim editor. The language it’s written in is C. The editor is running under the X Windows system, using the Linux operating system, both of which are written in C. In fact the majority of the applications I run on Linux are in C.

So why is C around when there are so many other languages around out there such as Java, Perl, ADA, C++, and others? The answer lies in the design and history of the language.

C was created in the 1960’s by Ken Thompson. The design process of this language is extremely interesting. Ken Thompson and his colleges wanted to use an old PDP-7 at Bell Laboratories. They needed a language which could be used to write an operating system in. So the language had to be very flexible.

The language was also designed to be as simple as possible. This wasn’t a formal requirement, but instead Mr. Thompson didn’t want to any more work than what was required.

As important as it is to look at what went into the language, it’s important to see what was left out. At the time, software was not being sold, so no one from marketing had any input into the design or implementation of the language.

It should be noted that at the time a lot of research was going into the theory of what makes a good programming language. Many different competing designs out there all trying to maximize the reliability and effectiveness of the programming language. Let’s face it, this was a time when a structured language such as ALGOL was considered a great innovation.

But Ken Thompson didn’t apply all this research to the design of his programming language. Instead he used old, proven syntax which he had seen and used in other languages. (Notable B and BCPL.) The result was a very solid language definition.

The result was the Ken Thompson designed a language to solve a problem that he had. Because Mr. Thompson was both the man with the problem and the person creating the solution, he was able to come with a very effective way of creating worked.

It’s important to understand that C was designed not by academics, who know all about theory, but little about the real world, not by a marketing department, who knows all about making a buck, but by someone who wanted to get a job done and who knew how to do it.

C did the job and did it very well. In fact C does a variety of jobs and does most of the well. Because of that and because there were no good alternatives available, the C language grew until it became the biggest language of the world.

But that was in the past. What about today? Aren’t there better languages out there? What about C++ and Java.

C++ is an object oriented language grafted on to a procedural language, C. It does add objects to the C and most people think that objects are a good thing. So why is it not being used more?

The answer is that although C++ is better than C, it’s not that much better. In other words, the cost of converting C programs to C++ out weighs the benefit of conversion. Another problem is that there are not enough C++ programmers out there. So many companies faced with a huge amount of C code and a large number of C programmers, tend to stay with C. That’s not to say that C++ won’t grow in popularity in the future, but for now there’s still a lot of C out there.

Now let's talk about the new "hot" language Java. This language was designed by the people at Sun to help them sell computers. Java was designed to work in an environment where you had a main server and a lot of little computers (called thin clients) on peoples desk. All the program and files would be kept on the server and everyone would use a small, cheap compute engine to do their work.

Trouble was that people liked their personal computers and didn't want to go back to the of the old mainframe. That's not to say that the people who designed Java weren't pretty smart. They were. It just that they designed a language to solve a problem that didn't exist. So Java became a language in search of home and found in the web browser. But it's never been more than a cult language outside this market. (I know I'll get flamed for saying that, but for every person who points out that they are using Java to create a real application, I can point to thousands who are developing in C. Java just isn't there yet.)

There more modern language I haven't discuses yet, Perl. Perl is the language of the web and the system administrator. That's because its great for writing string oriented programming. For example, the basic business program is:

```
1.Read it.
2.Munch it. (i.e. Do some data processing on it)
3.Sort it.
4.Write a report.
```

Perl is also idea for writing web based applications which use databases. But Perl and C attack a different problem set. Perl is very high level and string oriented. C is close to the hardware and number oriented. Perl is slow, C is fast.

Finally, there are things you can do in C that you can't do in Perl. Remember the Perl interpreter is written in C.

So for now at least, C is the language of choice for most programming applications. In the new few months we'll take a closer look at this language and learn more about it.

Problem #1 – Hello World

In this section we will present you with a monthly puzzle to test your knowledge of C.

Since most C text books start with "Hello World", we decided that we would too. Only ours is broken. Can you tell why?

```
1 #include <stdio.h>
2 void main()
3 {
4     printf("Hello World!\n");
5 }
```

Hello World — Hint #1

This program is interesting in that it compile perfectly on all known compilers, yet it does not conform to the standard.

That brings up an interesting question: What's is a standard anyway? Is it what the standards committee decides or is it what everyone does?

Hello World — Hint #2

What is the exit code returned to the operating system?

(Go the page bottom for the answer.)

About the Author Steve Oualline wrote his first program when he was eleven. It had a bug in it. Since that time he has studied practical ways of writing programs so that the risk of generating a bug is reduced. He has worked for Motorola and Celerity Computing, and is currently a special consultant for Hewlett Packard, working in the research department of their Ink-Jet division. He can be reached by oualline@ftp.oualline.com

Hello World — Answer

According to the C standard function main returns an int. It is not a void function. A void function returns a random exit code to the operating system. A properly written version of this program looks like:

```
#include <stdio.h>
int main()
{
    printf("Hello world!\n");
    return (0);
}
```