

Re: Quick question about streams...?

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.language/2006-05/msg00259.html>

- *From:* Tim Roberts <timr@xxxxxxxxx>
 - *Date:* Sat, 06 May 2006 16:38:07 -0700
-

Robby <Robby@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote:

Okay, I don't mind learning all there is to learn, but wait a minute...
All the commands that I am doing in Charles Petzold's book (for loops, structures, malloc, and Create file.... isn't that all "C" language)

No. For loops and structures are the C language. malloc is part of the C run-time library, available on every operating system that has a C compiler. CreateFile is a Win32 API. Windows-specific, unrelated to C.

These are important differences. Compilers don't care what functions you call. A C program can call run-time library functions, or Windows functions, or Linux functions, or VMS functions. Doesn't matter. It's all C. When you try to RUN the program, that's when you get into the libraries and the APIs. But, at that point, it isn't C or C++ any more, it's a compiled, machine-language binary.

And isn't the API a bunch of C functions?

No. The API is a bunch of functions. They happen to be defined based on the C calling conventions, but they can be called from any language that can provide that. Pascal, Visual Basic, C++, Forth, Python, Perl; there are many languages that can call the Win32 APIs.

I was always confused on this.... For once and for all.....Can I confirm something.

Our processor right runs binary code... right....

Processors run machine language. Raw bytes of binary that are executed by the processor.

Re: Quick question about streams...?

which in turn they developed assembler on top of binary code right....

"They"? It's possible that the first x86 assemblers were written in a higher-level language on some other processor. You'd assemble a program on that other computer, then copy it over to your x86 for execution.

In which then they invented C which runs on assembler right....

I don't know what "runs on assembler" means. A C compiler usually starts out as a little bootstrap program written in assembler; eventually, the bootstrap gets good enough so that the rest of the compiler can be written in C. Most C compilers create machine language directly, so that an assembler isn't involved. (The gcc compiler does create assembly code, and then calls an assembler to finish the job.)

Then they developed DOS on C right..... or wrong?

Originally, DOS was written in assembler. The x86 C compilers just weren't good enough at the time.

Whereby they made Windows out of C language right.... coupled with C functions as the API right or wrong... or am I way off?????

Windows was written in a mixture of C and hand-optimized assembler. Windows NT and its derivatives are almost entirely in C and C++.

And then how does C++ and VC++ tie into all of this????
And I guess mfc is built to run on the API right....

MFC is a large set of C++ wrappers that makes the API easier to use. Most MFC functions eventually call Win32 APIs.

And what about atl... what is that?

You will have to learn about C++ for that. ATL is a set of template classes that make Windows apps and components easier to write. ATL has less overhead than MFC.

and .net runs on the API also or on C++?

Re: Quick question about streams...?

Re: Quick question about streams...?

I don't know what you mean by "runs on". The core part of .NET is probably written in C or C++. Most of its large run-time library is probably written in C#.

And what's a Frameworks?

A framework is just a large library of procedures and functions to help you write some kind of application. MFC is a framework. The .NET common-language runtime is a framework.

--

- Tim Roberts, timr@xxxxxxxxxx
Providenza & Boekelheide, Inc.

.