Is it worth it to learn a dialect of assembly? [closed]

Asked 16 years, 4 months ago Modified 12 years, 7 months ago

Viewed 1k times

▲

4

▼

Closed. This question is opinion-based. It is not currently accepting answers.

Want to improve this question? Update the question so it can be answered with facts and citations by editing this post.

Closed 10 years ago.

Improve this question

My goals are focused on software application development, and maybe web application development, but most likely desktop applications. I'm embarking on a path to becoming more familiar with C/C++, but should I go much lower than that, into assembly? Or would I not have a benefit for my long-term goals?

assembly

Share edited May 2, 2012 at 12:17

Improve this question

Follow

kristianp
5,895 ● 39 ● 62

asked Aug 26, 2008 at 12:54

Thomas Owens
116k ● 99 ● 317 ● 436

11 Answers

Sorted by: Highest score (default)

▲

11

▼

I am really surprised to see so many "no" answers to this question. I think you *should* learn assembly.

I do not expect that you would ever *use* assembly directly as part of your job. But it does not follow from that that you should not learn it.

Learning assembly will teach you about what is going on inside the computer. It will help you to understand what the software is actually doing.

It's really about professionalism. Are you going to be a professional software engineer? Or are you going to be a copy-and-paste hack? Sure, the latter may pay the bills, but being a professional is so much more satisfying.

To hear someone say, "Nah, don't bother learning assembly," sounds to my ears like "Here's the cookbook for building bridges. You don't need to learn about physics or engineering to build a bridge. Just follow these recipes." No, thank you.

Share Improve this answer

Follow

answered Oct 6, 2008 at 1:36

Jeffrey L Whitledge
59.4k ● 9 ● 74 ● 100

3

G'day,

I learnt PDP assembler when I did my Elect. Eng. degree in the late '70's. The last dialect of assembler that I really used had four different modes of memory addressing. Last dialect I ooked at had 17 modes!

Not sure what learning assembler really gives you nowadays. Back then it was an essential part of a CS stream in my elect. eng. degree.

As to learning C++ I'd just sit down and work through "Accelerated C++" which approaches C++ in its own right and not as "C with other bits".

As to C, I'd just work through the latest version of "C Programming Lanuage" (a.k.a.) K'n'R

Hope this helps.

cheers, Rob

Now if you'd asked about nano-progrmming... (-:

Share Improve this answer

Follow

answered Aug 26, 2008 at 13:21

Rob Wells
37k ● 13 ● 84 ● 147

2

"Is it worth it to learn a dialect of assembly?"

I have programmed assembly professionally. M68k running a fax machine and scanner. Also windows VxDs (virtual device drivers) back in the windows 3/3.1 days before they had a real kernel.

When you code assembly to do ordinary software-y kinds of tasks (copying memory, concatenating strings, calling interrupt handlers etc), it's kind of interesting. Sometimes you code assembly to be called by C code to perform some specialized task as fast as possible on a given processor. That can be more interesting because you're looking for ways to take advantage of every cycle the processor gives you. You care about what's in the processor's L1 cache. You care about aligning data in memory to avoid cacheline hits (if I remember the term). You care about dual pipeline processor architecture and using the right 2 or 3 or 4 instructions in the right order so that 2 or 3 or 4 things are happening with a single clock tick (-one- of those HZ in the processor's XgHz).

When you code assembly to drive custom hardware, now you're doing stuff like filling a 16byte buffer of memory, setting up a DMA operation and firing that data over to a controller that is doing something like driving a laser printer drum. And the drum is turning and can't be stopped and wants its next 16 bytes within the next 5us. Of course that can be done in C or C++. But the examples are endless.

I would maybe trim off the last half of your question "Is it worth it to learn a dialect of assembly?" And make it "Is it worth it to learn?"

If you love programming, how you define "worth" involves some component of the love of programming. In that sense, I've never learned something in programming and not thought it was worth it. Even if I didn't use it much afterwords.

In that same sense, I would almost say that the harder something is to learn, the more "worth" it it is.

But all that fluffy crap aside, I believe it is worth it to at least get -some- assembly background. Go ahead and figure out how to write the assembly to replace a few simple stdlib routines like strcpy, memmove etc. Then try to optimize them, calling them from C a million times while timing it.

Share Improve this answer Follow

answered Oct 6, 2008 at 1:04

user126593

2,797 ● 4 ● 21 ● 16

▲

1

▼

It's probably not going to have a whole lot of benefit unless you have a direct application for it. If you're going for general knowledge, C/C++ is a fine place to start.

That said, the challenges that assembly poses are very interesting and it requires a pretty different mindset to get things done.

✓

I spent a little time learning Z80 assembly by programming the TI-86 calculator. The Z80 instruction set is pretty small and the novelty of programming a calculator in assembly is very amusing.

ticalc.org has a lot of good resources on TI assembly programming.

Share Improve this answer Follow

answered Aug 26, 2008 at 13:02

Mark Biek

151k ● 54 ● 158 ● 201

You can make homebrew Nintendo Gameboy games with (a subset) of Z80 assembly, as well. – James Feb 26, 2014 at 7:22

▲

1

▼

I wouldn't start learning ASM. If you want to learn C/C++ then start with that. As the quality of your code matures, you may find you have a need for ASM. 99% of the time you won't, but every now and then you might need it.

Also, it does help to know ASM in terms of understanding what C/C++ is doing behind the scenes. But again, until you get more advanced, you probably won't have a need for it.

Share Improve this answer Follow

answered Aug 26, 2008 at 12:57

Mark Ingram

73.5k ● 53 ● 177 ● 233

1

I did, and I think it helped me at the time. It doesn't help me day to day anymore, but I think it would depend on your job.

I learned assembler 20 years ago on a Commodore and again in University on an IBM mainframe. I can't say it helps me in my current job.

Share Improve this answer Follow

answered Aug 26, 2008 at 12:59

ScottStonehouse
24.9k ● 7 ● 33 ● 34

1

I agree with Mark. I think it's similar to learning MSIL when writing in C#, VB.NET, or another .NET language. It helps to know what's going on under the hood, but you could go your entire life creating applications that work and never need it.

Share Improve this answer Follow

answered Aug 26, 2008 at 13:05

Inisheer
20.8k ● 9 ● 54 ● 83

1

No. Unless you want to for fun, you really don't need learn assembly.

There are some things you need to know assembly for, like driver creation, OS development, exploit development, but aside from that, I personally believe you can quite happily code forever without knowing it.

If you do need to learn assembly, you'll know it - I wouldn't learn it for the sake of learning it..

Share Improve this answer Follow

answered Aug 26, 2008 at 13:08

dbr
169k ● 69 ● 283 ● 347

1

If you are writing unmanaged C++, it is occasionally invaluable to know at least basic x86 assembly, binary number systems, etc. I primarily do C/C++ development, and I occasionally need to debug production code for errors that are so specific to the machine-code representation produced by the compiler that the only way to find, and then fix the bug is to read the decompiled assembly and ascertain why the compiler generated it as such.

For more information on assembly, see the question: What is the best way to learn Assembly? Specifically, for someone who has experience in dynamic languages.

Share Improve this answer Follow

edited May 23, 2017 at 11:45

Community Bot
1 ● 1

answered Aug 26, 2008 at 13:14

Chris
6,842 ● 6 ● 53 ● 67

1

Assembly is not very difficult. Once you're familiar with C, spend a day or two learning basic assembly. Its helpfulness in terms of debugging is tremendous, plus its fun being able to write code that beats the C equivalent speed-wise by a factor of 10, 15, or more.

Share Improve this answer Follow

answered Oct 6, 2008 at 1:24

Dark Shikari
8,009 ● 4 ● 28 ● 38

0

I would not suggest to learn a "modern assembler language".

However knowing a little about MOS 6510 Assembler and browsing through the disassembled C64 Kernel, a.k.a. it's OS and BASIC Interpreter, helped me a lot to understand what's going on inside a computer - stuff like interrupts and memory pages.

This could possibly help you to give you hints how to write optimized code in other languages. A lot of that - however - is already done by modern compilers, so I'd only suggest that if you're interested in what's going on inside of that black box.

Share Improve this answer Follow

answered Aug 26, 2008 at 13:27

BlaM
28.8k ● 33 ● 93 ● 106