

Forum:[Hack Forums](#)

Thread: [Just started Assembly Language. I love it. Great language.](#)

RazerBlade - October 24, 2012, 2:50 pm

Just started Assembly Language. I love it. Great language. Not a troll. It's a great language seriously. People normally start out at high languages at Java or C++ etc. But it's kind of interesting to start low, then build up. Which is going to be my approach to coding.

lol. i know. I'm a fricken nooooooob.

aixeiger - October 24, 2012, 4:33 pm

Yeah! finally all code is assembly! but it's so extended! to study!

RazerBlade - October 24, 2012, 9:48 pm

Assembly is hands down my favorite language. This is absolutely no joke. Love learning it!

(° ? °) - October 24, 2012, 10:02 pm

ASM is really interesting when you start writing metamorphic or self-modifying code.

BaussHacker - October 25, 2012, 4:20 am

What assembler are you using, if I can ask? :D

0_0_ - October 25, 2012, 6:40 am

From what I hear, it completely brainfucks you and is harder than learning Advanced Chinese. I'm glad you're progressing well.

buyinsurance - October 25, 2012, 7:57 am

Good job! As someone who learned to code C++ first I am kind of struggling with assembly. It's not impossible by any means, but I think what makes it harder for myself is that I am studying about memory layout and things like that. You might want to look into those things when you can write a couple of programs.

Ac1dFr33z3 - October 25, 2012, 10:52 am

(Oct 24, 2012 02:50 PM)RazerBlade Wrote: [\[CITE\]](#) Just started Assembly Language. I love it. Great language. Not a troll. It's a great language seriously. People normally start out at high languages at Java or C++ etc. But it's kind of interesting to start low, then build up. Which is going to be my approach to coding.

lol. i know. I'm a fricken noob.

I've been planning on learning Assembly for a while now what are you using to learn if I can ask.

mareeg - October 26, 2012, 5:46 am

Had to learn it in uni last year. It is quite fun, but pretty hard, but such a brain teaser. I recommend you use MARS as your editor

jazzycool - October 26, 2012, 6:59 pm

I accept the fact, that yup it is a lovely language, although i started out with it, but then I left half not because it was boring but it was my distraction abilities which forced me to run out of it.

ChewyMoon - October 26, 2012, 7:02 pm

I would love to learn ASM, but I can't any tutorials. Considering buying a book about it.

RazerBlade - October 26, 2012, 7:52 pm

(Oct 25, 2012 10:52 AM)Ac1dFr33z3 Wrote: [\[CITE\]](#) I've been planning on learning Assembly for a while now what are you using to learn if I can ask.

bro. I'm starting with the LC-3 machine. lol. don't think u'll want to start off with that rofl. it's basically ASSEMBLY plus stripped down version. hahaha

Love2D - October 26, 2012, 9:11 pm

That may be, but C is much faster than ASM.

RazerBlade - October 26, 2012, 9:17 pm

(Oct 26, 2012 09:11 PM)Love2D Wrote: [\[CITE\]](#) That may be, but C is much faster than ASM.

not necessarily. if we're doing embed system where we need code that are closer to the actual hardware deep down, ASM is hands down faster.

Love2D - October 26, 2012, 10:08 pm

(Oct 26, 2012 09:17 PM)RazerBlade Wrote: [\[CITE\]](#) not necessarily. if we're doing embed system where we need code that are closer to the actual hardware deep down, ASM is hands down faster.

Assembly is terrible when coding embedded systems. Are you an embedded systems programmer?

Han. - October 27, 2012, 9:02 am

(Oct 25, 2012 06:40 AM>Hello Friends Wrote: [\[CITE\]](#) From what I hear, it completely brainfucks you and is harder than learning Advanced Chinese. I'm glad you're progressing well.

Don't have an opinion on it if you've never tried. The only people who say that it's ridiculously difficult are those like you, who've never tried it.

RazerBlade - October 27, 2012, 9:36 am

(Oct 27, 2012 09:02 AM)hancsu Wrote: [\[CITE\]](#) Don't have an opinion on it if you've never tried. The only people who say that it's ridiculously difficult are those like you, who've never tried it.

yea, and funny thing is, I know assembly AND chinese. How ironic.

so @HelloFriends. it isn't that hard if u actually try lol

captmicro - October 27, 2012, 11:29 am

(Oct 26, 2012 10:08 PM)Love2D Wrote: [\[CITE\]](#) Assembly is terrible when coding embedded systems. Are you an embedded systems programmer?

ASM is terrible, but sometimes REQUIRED on things like PIC's when you need to write cycle-exact code for things like VGA/Component video (impossible in C).

RazerBlade - October 27, 2012, 11:44 am

(Oct 27, 2012 11:29 AM)captmicro Wrote: [\[CITE\]](#) ASM is terrible, but sometimes REQUIRED on things like PIC's when you need to write cycle-exact code for things like VGA/Component video (impossible in C).

ASM isn't terrible. Yes compared to functionality it's shit. ok. slow. crap. gay. GOTCHA.

but if you want to learn exactly how the workings of code actually go about, ASM gives you that insight, and it's this insight that differentiates an ASM coder from let's say, a JAVA coder. Sure a java coder can make cooler programs with more efficiency, but when it comes to really getting to know the hardware, ASM has the upper hand.

Ac1dFr33z3 - October 29, 2012, 10:47 am

(Oct 26, 2012 07:52 PM)RazerBlade Wrote: [\[CITE\]](#) bro. I'm starting with the LC-3 machine. lol. don't think u'll want to start off with that rofl. it's basically ASSEMBLY plus stripped down version. hahaha
Haha I have some verrrry limited experience with it your using a simulator right? When I was learning it a little a few years ago I had a teacher have me learn by looking a software code in ASM with a HEX editor. Actaully hated It but did learn. ?

tinkerman2000 - October 30, 2012, 1:50 pm

I am learning ASM as we speak. I am thinking to write a tutorial about the basics. Since I already learned it. I am working on intermediate stuff.

I use NASM. What do you use OP?

Han. - October 31, 2012, 7:19 am

Some good points have been raised.

Many people think that Assembly is automatically faster than something like C. When all that matters is the code that is actually being written.

It's perfectly plausible to have a well-written program in C be a whole lot faster than a badly written program in ASM.

oprichni - October 31, 2012, 11:50 am

I'm not sure what people meant by "assembly is slower than C". I guess/hope most here already know that C compiles to assembly? Handwritten assembly can be slower if the programmer doesn't know what he's doing, but that's not the language's fault.

I'm pretty sure the first post making that claim was trolling. Unless by "slower" they meant development time, in which case C would be faster.

tinkerman2000 - October 31, 2012, 3:38 pm

(Oct 31, 2012 07:19 AM)hancsu Wrote: [\[CITE\]](#) Some good points have been raised.

Many people think that Assembly is automatically faster than something like C. When all that matters is the code that is actually being written.

It's perfectly plausible to have a well-written program in C be a whole lot faster than a badly written program in ASM.

I have seen benchmarks of a hello world console print of text. It showed ASM executes faster then a program written in C or C++.

I cannot tell you why since I am not a pro at ASM. I am learning ASM.

The only thing I can say is that I seen benchmarks and have had others explain to me why it's faster. They said it's more direct.

C and C++ gets converted to ASM which gets converted to OP code.

However, C and C++ has functions. C++ has pre-created functions and it loads that with your app. So since you may never need to use every function that C and C++ as in it's language. It will hog down the processing time and space. Where as ASM allows you to directly talk to the hardware. You can write functions that gets converted to OP code. It makes your app smaller and run faster.

Any other language even C and C++ all have been created by the use of ASM. ASM language is

can execute faster and have less space used but takes longer to develop. The only reason you can directly talk to the hardware without any api's or libs.

I know you can do system calls to the linux Kernal. I was told you can do the same with windows... however, others told me that is a load of BS. they said that windows in 32bit mode is considered to be in protected mode. Which means you cannot make calls to the windows Kernal. You need to use their api.

(Oct 31, 2012 07:19 AM)hancsu Wrote: [\[CITE\]](#) Some good points have been raised.

Many people think that Assembly is automatically faster than something like C. When all that matters is the code that is actually being written.

It's perfectly plausible to have a well-written program in C be a whole lot faster than a badly written program in ASM.

I think what they meant is that you can make applications that can directly go to the source bypassing libs and other code that you may not have to use all of it.

For instance lets say that I want to draw on the screen a sky that has clouds floating from right to left.

You can use a graphics engine which can do this for you. However, you may not use many of the graphics if it's a 3d and a 2d graphics engine. You would use the 2d functions. the 3d won't be used but you will still have to store that and still load that into ram.

__init__ - November 1, 2012, 2:53 am

I've just had a look at an example source code and it looks CRAZY hard. When I look at C++ or something, I can kinda work out what something does, but I have no idea what any of that does xD

Han. - November 1, 2012, 4:13 am

(Nov 1, 2012 02:53 AM)TechnoHippie Wrote: [\[CITE\]](#) I've just had a look at an example source code and it looks CRAZY hard. When I look at C++ or something, I can kinda work out what something does, but I have no idea what any of that does xD

Then learn.

You don't get anywhere by looking at something, thinking it's too hard and then walking away ?

(Oct 31, 2012 03:38 PM)tinkerman2000 Wrote: [\[CITE\]](#) I have seen benchmarks of a hello world console print of text. It showed ASM executes faster then a program written in C or C++.

I cannot tell you why since I am not a pro at ASM. I am learning ASM.

The only thing I can say is that I seen benchmarks and have had others explain to me why it's faster. They said it's more direct.

C and C++ gets converted to ASM which gets converted to OP code.

However, C and C++ has functions. C++ has pre-created functions and it loads that with your app. So since you may never need to use every function that C and C++ as in it's language. It will hog down the processing time and space. Where as ASM allows you to directly talk to the hardware. You can write functions that gets converted to OP code. It makes your app smaller and run faster.

Any other language even C and C++ all have been created by the use of ASM. ASM language is can execute faster and have less space used but takes longer to develop. The only reason you can directly talk to the hardware without any api's or libs.

I know you can do system calls to the linux Kernal. I was told you can do the same with windows... however, others told me that is a load of BS. they said that windows in 32bit mode is considered to be in protected mode. Which means you cannot make calls to the windows Kernal. You need to use their api.

I think what they meant is that you can make applications that can directly go to the source bypassing libs and other code that you may not have to use all of it.

For instance lets say that I want to draw on the screen a sky that has clouds floating from right to left.

You can use a graphics engine which can do this for you. However, you may not use many of the graphics if it's a 3d and a 2d graphics engine. You would use the 2d functions. the 3d won't be used but you will still have to store that and still load that into ram.

Again, some very valid points.

However, a Hello World in Assembly is almost certainly going to be faster.

My point was that once you get into programs that are industry sized and being used for a company, if the programmer hired is awful at Assembly but a guru at C, then the C code will be good and the Assembly code will be bad, which in some cases could lead the the C code actually executing faster than the Assembly.

Obviously the programs that would apply to would have to be big enough so that it actually had some differences to it because you can't do everything in C the same as in Assembly, you have to find different ways.

D3vildog - November 1, 2012, 4:17 am

I'm learning reverse engineering and it's pretty fascinating to me.

I've done MASM32 projects in the past.

Han. - November 1, 2012, 4:17 am

(Oct 31, 2012 11:50 AM)oprchniki Wrote: [\[CITE\]](#) I'm not sure what people meant by "assembly is slower than C". I guess/hope most here already know that C compiles to assembly? Handwritten assembly can be slower if the programmer doesn't know what he's doing, but that's not the language's fault.

I'm pretty sure the first post making that claim was trolling. Unless by "slower" they meant development time, in which case C would be faster.

Yes, I know that C is compiled into Assembly which in turn is converted into machine code.

My point was that Assembly can be slower if the programmer is inefficient in the way he writes the code.

I didn't mean to come off as if I was saying that it was the languages fault, it is most definately the programmers fault.

__init__ - November 1, 2012, 6:40 am

(Nov 1, 2012 04:13 AM)hancsu Wrote: [\[CITE\]](#) Then learn.

You don't get anywhere by looking at something, thinking it's too hard and then walking away ?

Again, some very valid points.

However, a Hello World in Assembly is almost certainly going to be faster.

My point was that once you get into programs that are industry sized and being used for a company, if the programmer hired is awful at Assembly but a guru at C, then the C code will be good and the Assembly code will be bad, which in some cases could lead the the C code actually executing faster than the Assembly.

Obviously the programs that would apply to would have to be big enough so that it actually had some differences to it because you can't do everything in C the same as in Assembly, you have to find different ways.

Yeah I know, I'm learning Java atm though. I don't really see the need for me to learn ASM too.

K4Fr - November 1, 2012, 7:57 am

In general, for the same algorithm, C will almost always beat a good assembly programmer even though good assembly programmer can always either be equivalent to or beat C. To put that another way; a good assembly programmer could do better than a compiler but will choose not to...

tinkerman2000 - November 3, 2012, 8:44 am

(Nov 1, 2012 04:13 AM)hancsu Wrote: [\[CITE\]](#) Then learn.

You don't get anywhere by looking at something, thinking it's too hard and then walking away ?

Again, some very valid points.

However, a Hello World in Assembly is almost certainly going to be faster.

My point was that once you get into programs that are industry sized and being used for a company, if the programmer hired is awful at Assembly but a guru at C, then the C code will be good and the Assembly code will be bad, which in some cases could lead the the C code actually executing faster than the Assembly.

Obviously the programs that would apply to would have to be big enough so that it actually had some differences to it because you can't do everything in C the same as in Assembly, you have to find different ways.

Ya, I agree with you that it depends on the programmer and his good habits to write less code as possible and be efficient.

Just saying that the ASM language itself can give you that speed edge if you follow good programming practices. The language itself will make the program faster however if you lets say use Windows 32 api to draw the windows and menus then that be the same speed as C. Since the code is the same. Their might be a small speed difference with ASM. It could execute your ASM commands faster then C commands. So not talking about the API but the code where you utilise the api. That could would be faster but not by much and it won't make a significant change in speed. Nothing that is noticeable.

I am just saying that ASM can make your programs faster but that's only if you use ASM to right the code more rawly or towards the o ring levels. AKA the more direct code.

ASM gets translated to op codes which is the most direct programming you can do towards hardware. So for like an OS if you can do system calls to the OS's kernel that will be quicker in

speed to execute then using an API that does the same thing but has a lot of fuss on top of it.

AlexChambers - November 3, 2012, 9:41 am

hope you enjoy doing this mate good thing to do :)

SmG R3c0N - November 12, 2012, 3:57 am

<http://ziplib.com/emu8086/>

I recommend emu8086. I'm not an expert but its a 8086 emulator with a bunch of features.

Not sure if doing assembly just for 8086 may be good in the longterm, but I think its nice for starters.

Its not free, either purchase or find it. ;D