

Forum:[Hack Forums](#)

Thread: [\[Assembly\] I'll be taking a computer organization and architecture class.](#)

Lindros - November 6, 2019, 8:55 pm

It will be in assembly. The only language I have had exposure to so far is Python.

Anything I should do to prepare? How would you rate the difficulty. I saw a tiny bit of a YouTube tutorial just to get some exposure to the information before hand. It looks kind of intimidating at first. Is there still some real uses for assembly? If so, what is it good for?

Fill me in.

SIM-3364521 - November 7, 2019, 10:25 am

actually for me personally asm (8085) was easier to pick up than python tbh. But that really depends on the person.

and ofc. there are still uses for it:

- micro controller programming
- binary exploitation
- reverse engineering and general debugging

also i do find my self declaring asm blocks in c++ as well for small tasks.

Especially when i want that code exactly how it is in the memory or to further optimize performance critical parts of the program.

The c++ compiler does a good job at that already but can also mess stuff up.

Often beause it doesnt know the context.

also i think its really needed to get a low level understanding of how programs run and do stuffs to become a better programmer over all.

also although im sure assembler knowledge would help you in a computer architecture class they often stay one step closer to the actual hardware and dont go in depth about assembler.

Its often more about how an instruction gets executed on the hardware. And how different architecture types like CISC or RISC handle more complex instructions and how their assembler language looks like (as an overview without goooing too deep into assembler nor the transistor level part)

If that makes sence ^^

Lindros - November 7, 2019, 11:07 am

(Nov 7, 2019 11:25 AM)Erarnitox Wrote: [\[CITE\]](#) actually for me personally asm (8085) was easier to pick up than python tbh. But that really depends on the person.

and ofc. there are still uses for it:

- micro controller programming
- binary exploitation
- reverse engineering and general debugging

also i do find my self declaring asm blocks in c++ as well for small tasks.

Especially when i want that code exactly how it is in the memory or to further optimize performance critical parts of the program.

The c++ compiler does a good job at that already but can also mess stuff up.

Often because it doesn't know the context.

also i think its really needed to get a low level understanding of how programs run and do stuffs to become a better programmer over all.

also although im sure assembler knowledge would help you in a computer architecture class they often stay one step closer to the actual hardware and don't go in depth about assembler.

Its often more about how an instruction gets executed on the hardware. And how different architecture types like CISC or RISC handle more complex instructions and how their assembler language looks like (as an overview without going too deep into assembler nor the transistor level part)

If that makes sense ^^

Yeah that does make sense, thank you for the reply. I'm excited to learn.

SIM-3991524 - November 19, 2019, 5:25 pm

You'll want to find out what assembler you'll be writing in. wikipedia article "Comparison of assemblers". Most common are MASM, NASM, FASM. In college my professor had his own assembler that he used to teach the class, so who knows

Loki123 - December 5, 2019, 3:34 am

(Nov 7, 2019 11:25 AM)Erarnitox Wrote: [CITE](#) actually for me personally asm (8085) was easier to pick up than python tbh. But that really depends on the person.

and ofc. there are still uses for it:

- micro controller programming
- binary exploitation
- reverse engineering and general debugging

also i do find myself declaring asm blocks in c++ as well for small tasks.

Especially when i want that code exactly how it is in the memory or to further optimize performance critical parts of the program.

The c++ compiler does a good job at that already but can also mess stuff up.

Often because it doesn't know the context.

also i think its really needed to get a low level understanding of how programs run and do stuffs to become a better programmer over all.

also although im sure assembler knowledge would help you in a computer architecture class they often stay one step closer to the actual hardware and don't go in depth about assembler.

Its often more about how an instruction gets executed on the hardware. And how different architecture types like CISC or RISC handle more complex instructions and how their assembler language looks like (as an overview without going too deep into assembler nor the transistor level part)

If that makes sense ^^

I was in a similar class this semester and this reply is spot on.

The assembler came up maybe once or twice when we talked about what happens when we compile a C program. In my school we didn't do any programming in Assembly but had to learn how to read Assembly code and reverse engineer basic C programs from assembly. Follow what happens to loops and how we should convert a simple while loop into do-while loop and from there into a "goto" program which then is pretty identical to what happens in Assembly.

We went into optimizing our code as, for example, calculating $x*y$ where x and y is a constant can

be very expensive if done every time inside a loop. Seeing how the compiler both fixes that and places the calculations outside of the loop and while it might not do it and how we can optimize our code so we aren't wasting resources.

But what OP should do to prepare himself for that class he is taking is really hard to say. It all depends on how the class is based but my perspective would be to work a little with C and basic C applications might help.