

Forum:[Hack Forums](#)

Thread: [what's the best coding language for hacking](#)

Infamia - September 23, 2018, 2:04 pm

i'm dead curious lmk

is python any good?
should i learn c or c++
x86 or arm{v7}?
html > css?
js > html?
js > html > python?

i'm so confused

which one of these programming languages are better guys?

Faith Connors - September 23, 2018, 2:09 pm

haskell or emacs lisp
prove me wrong protip you cant

jokes aside, i feel that risc-v should start taking off soon for consumer grade, probably within the next two years so getting a headstart on that would be a good choice

Infamia - September 23, 2018, 2:57 pm

(Sep 23, 2018 02:09 PM)Faith Connors Wrote: [\[CITE\]](#) haskell or emacs lisp
prove me wrong protip you cant

jokes aside, i feel that risc-v should start taking off soon for consumer grade, probably within the next two years so getting a headstart on that would be a good choice
sorry for taking too long to respond i was mentoring somebody on how to become a hacker (related thread)

haskell is such a good language tbh [lisp too, but haskell is just <3 <3 <3]

so i took a quick summary read [here](#) and it describes the ISA as an industry-grade, extensible and a practical paradise for further implementation upon certain subarchitectures and use-cases.
appears it's like some visual studio c# form designer, except for ISAs, sounds good, i guess, i would read further into it but i'm already six feet deep into x86, mathematics and hair and beauty, learning some other completely unrelated ISA would seem like a handful.
are you interested into getting into it?

Faith Connors - September 23, 2018, 6:54 pm

(Sep 23, 2018 02:57 PM)Infamia Wrote: [\[CITE\]](#) sorry for taking too long to respond i was mentoring somebody on how to become a hacker (related thread)

haskell is such a good language tbh [lisp too, but haskell is just <3 <3 <3]

so i took a quick summary read [here](#) and it describes the ISA as an industry-grade, extensible and

a practical paradise for further implementation upon certain subarchitectures and use-cases.

appears it's like some visual studio c# form designer, except for ISAs, sounds good, i guess, i would read further into it but i'm already six feet deep into x86, mathematics and hair and beauty, learning some other completely unrelated ISA would seem like a handful.

are you interested into getting into it?

hi also sry for late response

im very interested since its likely the next frontier after x86 (or openpiton is the only real competitor due to scalability) but im currently trying to learn bout x86 as well so im caught up in that. not six feet deep tho, probs only one foot rn since im taking it slow and focusing on studies.

maybe next year if i get a handle on stuff more lol

Infamia - September 24, 2018, 5:53 am

(Sep 23, 2018 06:54 PM)Faith Connors Wrote: [\[CITE\]](#) hi also sry for late response

im very interested since its likely the next frontier after x86 (or openpiton is the only real competitor due to scalability) but im currently trying to learn bout x86 as well so im caught up in that. not six feet deep tho, probs only one foot rn since im taking it slow and focusing on studies.

maybe next year if i get a handle on stuff more lol

hi sry for late response

Quote:OpenPiton is the world's first open source, general-purpose, multi-threaded many-core processor

i have a feeling openpiton is going to be a fairly competitive project based off the fact that it was developed first

Quote:OpenPiton is scalable and portable; the architecture supports addressing for up to 500-million cores, supports shared memory both within a chip and across multiple chips, and has been designed to easily enable high performance 1000+ core microprocessors and beyond

yh rip risc-v, it seems like openpiton is targetted at both consumers and people in industry whereas risc-v appears more industry-prevalent.

idk tho i'm just taking short scans of the papers published and comparing thick-lined details.

is x86 easy for u? i never really bothered to overview all the concepts and theoretical parts of x86 and just read 25% of a book here and there and then started reading code and reverse engineering things, it's hard to get used to when every other program is using a different calling convention and memory models, and some programs are obfuscated and just plain autism.

Faith Connors - September 24, 2018, 6:32 am

(Sep 24, 2018 05:53 AM)Infamia Wrote: [\[CITE\]](#) hi sry for late response

i have a feeling openpiton is going to be a fairly competitive project based off the fact that it was developed first

yh rip risc-v, it seems like openpiton is targetted at both consumers and people in industry whereas risc-v appears more industry-prevalent.

idk tho i'm just taking short scans of the papers published and comparing thick-lined details.

is x86 easy for u? i never really bothered to overview all the concepts and theoretical parts of x86 and just read 25% of a book here and there and then started reading code and reverse engineering things, it's hard to get used to when every other program is using a different calling convention and memory models, and some programs are obfuscated and just plain autism.

thing is, as it stands, getting openpiton hardware is impossible for the regular developer unless they want to get into computer engineering and build their own shit.

see their tutorial docs:

https://parallel.princeton.edu/openpiton/tutorial_slides/openpiton-isca16-opening.pdf

you're either building your own fpga or using a board that'll cost you 160\$ on top of whatever else you need. risc-v on the other hand is more readily available with their qemu support and web-based simulator/interpreter (or a 125\$ board if you're going native). so as a result, i think it'll take off before

openpiton does, but i havent gotten my hands on either yet so i cant say for sure.

its easy in the sense that i can at least understand most code if its presented to me. like, i can follow through its whole path while keeping track of shit (for the most part). but if it ever comes down to building anything, i'm god-awful with it. but i found a rly nice resource thats been helping me get a better handle on things: http://www.sizecoding.org/wiki/Main_Page

Infamia - September 24, 2018, 6:49 am

(Sep 24, 2018 06:32 AM)Faith Connors Wrote: [\[CITE\]](#) thing is, as it stands, getting openpiton hardware is impossible for the regular developer unless they want to get into computer engineering and build their own shit.

see their tutorial docs:

https://parallel.princeton.edu/openpiton/tutorial_slides/openpiton-isca16-opening.pdf

you're either building your own fpga or using a board that'll cost you 160\$ on top of whatever else you need. risc-v on the other hand is more readily available with their qemu support and web-based simulator/interpreter (or a 125\$ board if you're going native). so as a result, i think it'll take off before openpiton does, but i havent gotten my hands on either yet so i cant say for sure.

its easy in the sense that i can at least understand most code if its presented to me. like, i can follow through its whole path while keeping track of shit (for the most part). but if it ever comes down to building anything, i'm god-awful with it. but i found a rly nice resource thats been helping me get a better handle on things: http://www.sizecoding.org/wiki/Main_Page

so your reason that openpiton is not going to take off as fast in consumer-grade markets is because it's much more expensive to acquire, experiment with and so forth?

<http://www.princeton.edu/~cloud/openpiton/>

can't you download one of the virtualmachine images from here and just experiment around with it like that?

i'll try it right now so either by the time you respond calling me an idiot i'll have finished the download or i'll cancel it.

<https://www.begin.re/> i heard this website is good for introductions into x86 and subsequently RE, it's my path i'm taking with x86 anyway since i'll learn about x86 and how people hide their secrets in programs all at the same time, better than just focusing on x86 and understanding RE-related concepts later, though that sounds better if i don't want to struggle in the long run i guess, i'll take a look at ur website tho k

Faith Connors - September 24, 2018, 6:54 am

(Sep 24, 2018 06:49 AM)Infamia Wrote: [\[CITE\]](#) so your reason that openpiton is not going to take off as fast in consumer-grade markets is because it's much more expensive to acquire, experiment with and so forth?

<http://www.princeton.edu/~cloud/openpiton/>

can't you download one of the virtualmachine images from here and just experiment around with it like that?

i'll try it right now so either by the time you respond calling me an idiot i'll have finished the download or i'll cancel it.

<https://www.begin.re/> i heard this website is good for introductions into x86 and subsequently RE, it's my path i'm taking with x86 anyway since i'll learn about x86 and how people hide their secrets in programs all at the same time, better than just focusing on x86 and understanding RE-related concepts later, though that sounds better if i don't want to struggle in the long run i guess, i'll take a look at ur website tho k

ive actually not seen any of that openpiton directory haha, i was mainly basing everything off their google group and the tutorial slides which tend to focus on the more theoretical or native side of things. if this vm image works then i'll probably start looking into it more seriously.

and ya, the site i listed isnt for re, but just generally using asm for extremely tiny programs (generally under 256b, used lots in the scene like demos n shit). been helping me get a better feel for finding different ways to do something, since im usually pretty garbage at optimizing code on my own lol

Infamia - September 24, 2018, 7:14 am

(Sep 24, 2018 06:54 AM)Faith Connors Wrote: [\[CITE\]](#) ive actually not seen any of that openpiton directory haha, i was mainly basing everything off their google group and the tutorial slides which tend to focus on the more theoretical or native side of things. if this vm image works then i'll probably start looking into it more seriously.

and ya, the site i listed isnt for re, but just generally using asm for extremely tiny programs (generally under 256b, used lots in the scene like demos n shit). been helping me get a better feel for finding different ways to do something, since im usually pretty garbage at optimizing code on my own lol

3 hours for 5.2gb not looking like my internet just got downgraded from shit to shit for torrenting

took a look at the examples on the website for getting started and i start wondering how i can even understand assembly since reading assembly by without any tricks added is a challenge but when there are some implicit tricks going on that's just a nono for me i can't handle my brain bursts and i start asking questions then i end up on wikipedia learning about the history of aarch64

plus it's windows related so it's bound to be on some gay, i don't really like websites that don't explain things well either:

Quote:The boot sector is loaded at 0000:7C00

why?

Quote:The stack pointer is 512 bytes beyond the end of the boot sector, so SP is likely 7E00h

why is it only likely to be around 0x7e00

http://www.sizecoding.org/wiki/General_Coding_Tricks @ Data is code, code is data

where would this even be used it seems like a completely premature optimization solely for fucking with people trying to read your code

its like the entire concept of sizecoding is meant for environments where (a) nobody is going to read ur code (b) you're working in under 510 bytes © you're trying to understand some other retard's code that followed the same tips & tricks (d) ur on that gay