

Forum:[Hack Forums](#)

Thread: [Are there any advantages to game hacking with C/C++ compared to Java/C#?](#)

Owen GB - January 12, 2016, 4:46 am

Title says all, sorry if this is a frequently asked question. Off the top of my head I can only think C++ is a more used language thus it's easier to create complicated game modifications / hacks.

Remix - January 12, 2016, 5:39 am

You can do hacks with C# as well but as far as I know you aren't to make them with Java

sqsgkrldwg - January 12, 2016, 5:40 am

C++ is not the most used language today.

If you want game hacking for yourself only then go for the language you prefer.

If you want to distribute cross-platform hacks you'll be smarter using Java, if ever you can hack with it.

All of those languages can do the trick externally.

If you inject a dynamic lib you must use the same language as the game was made with.

Owen GB - January 12, 2016, 6:16 am

(Jan 12, 2016 06:40 AM)Lef fe Wrote: [\[CITE\]](#) C++ is not the most used language today.

If you want game hacking for yourself only then go for the language you prefer.

If you want to distribute cross-platform hacks you'll be smarter using Java, if ever you can hack with it.

All of those languages can do the trick externally.

If you inject a dynamic lib you must use the same language as the game was made with.

Thanks for the info. Fyi, most hacks I've saw and came across are written with C++, hence why I mentioned that. But I have little to no knowledge of memory injection and so on, so I'll leave that to the pros.

WojeT - January 12, 2016, 6:26 am

You will probably find more examples and information using C++ than other languages.

Remix - January 12, 2016, 6:44 am

(Jan 12, 2016 06:40 AM)Lef fe Wrote: [\[CITE\]](#) C++ is not the most used language today.

If you want game hacking for yourself only then go for the language you prefer.

If you want to distribute cross-platform hacks you'll be smarter using Java, if ever you can hack with it.

All of those languages can do the trick externally.

If you inject a dynamic lib you must use the same language as the game was made with.

People are still implying that Java is the "best" cross-platform language

It's not

Owen GB - January 12, 2016, 6:50 am

(Jan 12, 2016 07:44 AM)+Remix Wrote: [\[CITE\]](#) People are still implying that Java is the "best" cross-platform language
It's not
Maybe not the best, but imo the easiest to use.

Remix - January 12, 2016, 6:57 am

(Jan 12, 2016 07:50 AM)Owen GB Wrote: [\[CITE\]](#) Maybe not the best, but imo the easiest to use.
If you just care for the difficulty and not the quality visit the [AutoIT forum](#)

sqsgkrldwg - January 12, 2016, 7:05 am

(Jan 12, 2016 07:44 AM)+Remix Wrote: [\[CITE\]](#) People are still implying that Java is the "best" cross-platform language
It's not
It's managed code. The builds will works on every systems that have the tools to run Java.
The library is also way more full than the standard C++.

We can't even compare C++ and Java because C++ is not managed ...

I prefer writing everything from sratch and losing time, but you and I can admit Java is a good cross-platform language ... because it is mainstreamed and easy to use.
I personally prefer Python over Java. But Python is a scripting language, and looks like less lambda-user-oriented than Java.

We can compare C# and Java, they are both managed. But C# is just made for Windows (Though I read somewhere they are now distributing it open-source, not sure ... I don't want to say bullshit ...)

What is your preferred cross-platform language ?

Owen GB - January 12, 2016, 7:23 am

(Jan 12, 2016 07:57 AM)+Remix Wrote: [\[CITE\]](#) If you just care for the difficulty and not the quality visit the [AutoIT forum](#)

I don't care for difficulty. The fact is most thing's I create for an advantage I create for myself, and only myself. So cross platform compatability isn't an issue for me. If I was to create a public program, then I'd care about it.

(Jan 12, 2016 08:05 AM)Lef fe Wrote: [\[CITE\]](#) It's managed code. The builds will works on every systems that have the tools to run Java.
The library is also way more full than the standard C++.

We can't even compare C++ and Java because C++ is not managed ...

I prefer writing everything from sratch and losing time, but you and I can admit Java is a good cross-platform language ... because it is mainstreamed and easy to use.
I personally prefer Python over Java. But Python is a scripting language, and looks like less lambda-user-oriented than Java.

We can compare C# and Java, they are both managed. But C# is just made for Windows (Though I read somewhere they are now distributing it open-source, not sure ... I don't want to say bullshit ...)

What is your preferred cross-platform language ?

Java. Especially for beginners. It's easy to understand and mature to.

Remix - January 12, 2016, 7:39 am

(Jan 12, 2016 08:05 AM)Leffe Wrote: [\[CITE\]](#) It's managed code. The builds will work on every system that has the tools to run Java.

The library is also way more full than the standard C++.

We can't even compare C++ and Java because C++ is not managed ...

I prefer writing everything from scratch and losing time, but you and I can admit Java is a good cross-platform language ... because it is mainstreamed and easy to use.

I personally prefer Python over Java. But Python is a scripting language, and looks like less lambda-user-oriented than Java.

We can compare C# and Java, they are both managed. But C# is just made for Windows (Though I read somewhere they are now distributing it open-source, not sure ... I don't want to say bullshit ...)

What is your preferred cross-platform language ?

???

Python is an interpreted programming language

At least python developers claim it

Java would be a better cross-platform language without the shitty vm

WojeT - January 12, 2016, 8:02 am

(Jan 12, 2016 08:39 AM)+Remix Wrote: [\[CITE\]](#) ???

Python is an interpreted programming language

At least python developers claim it

Java would be a better cross-platform language without the shitty vm Without the VM you would need to compile it to machine code, and to get platform "independence" you would need to compile it once for each platform you want to run it on. At that point you might as well just program in C, you won't find a language that has support for more platforms than C.

Remix - January 12, 2016, 9:14 am

(Jan 12, 2016 09:02 AM)WojeT Wrote: [\[CITE\]](#) Without the VM you would need to compile it to machine code, and to get platform "independence" you would need to compile it once for each platform you want to run it on. At that point you might as well just program in C, you won't find a language that has support for more platforms than C.

You know that cross-platform compatibility isn't the most important part?

I meant that the performance is bullshit

Narvikz - January 12, 2016, 1:29 pm

What I have read above is mostly bullshit to be honest.

You want to stick to the same language as the game you're programming for and (preferably) the same compiler even.

This will help avoid any incompatibilities that could happen if done otherwise.

So, CS:GO for example, is coded in C++ and compiled using the Microsoft C++ Compiler, by sticking with it you're making it so much easier for your work, since the calling conventions basically are the same which makes it much easier to hook stuff without having to deal with naked hooks and shitty inline function prologues and epilogues.

Really, there's no reason AT ALL to be using inline asm in 2016, please...

It's REALLY REALLY easy to hook shit like that.

Just hook `_stdcall` as `_stdcall` by replacing the first 5 bytes (or 6-7-8, whatever, depending on instructions placed there) with `JMP + address`, start your hook by the instructions that were replaced and end your hook by a `JMP` back to the place you hooked, no inline ASM, no nothing.

Same thing for `_thiscall` but now with `_fastcall` with `ecx` and `edx` pointers and arguments right to left once again after that.

In the end it would be WAY harder to use a C# module for this since the process wouldn't even be able to run the module without some hard work around it, even if you did it hooking stuff would be a pain in the ass really and there's really no benefit to it.

Now, Java? Wtf? Really?

sqsgkrldwg - January 12, 2016, 1:45 pm

(Jan 12, 2016 02:29 PM)Narvikz Wrote: [\[CITE\]](#) What I have read above is mostly bullshit to be honest.

You want to stick to the same language as the game you're programming for and (preferably) the same compiler even.

This will help avoid any incompatibilities that could happen if done otherwise.

So, CS:GO for example, is coded in C++ and compiled using the Microsoft C++ Compiler, by sticking with it you're making it so much easier for your work, since the calling conventions basically are the same which makes it much easier to hook stuff without having to deal with naked hooks and shitty inline function prologues and epilogues.

Really, there's no reason AT ALL to be using inline asm in 2016, please...

It's REALLY REALLY easy to hook shit like that.

Just hook `_stdcall` as `_stdcall` by replacing the first 5 bytes (or 6-7-8, whatever, depending on instructions placed there) with `JMP + address`, start your hook by the instructions that were replaced and end your hook by a `JMP` back to the place you hooked, no inline ASM, no nothing.

Same thing for `_thiscall` but now with `_fastcall` with `ecx` and `edx` pointers and arguments right to left once again after that.

In the end it would be WAY harder to use a C# module for this since the process wouldn't even be able to run the module without some hard work around it, even if you did it hooking stuff would be a pain in the ass really and there's really no benefit to it.

Now, Java? Wtf? Really?

Did you just post on the wrong thread ? ... ?

Who is talking about CS:GO ? What if OP just want to hack a browser-game ?

Who ever wrote ASM on this thread ?

I don't support C# but I can bet there is open-source projects to hook process functions with it.

Don't blame Java, you and I said it : He have to stick to the language of the game.
But we don't even know the target game in the first place !

Narvikz - January 12, 2016, 2:07 pm

(Jan 12, 2016 02:45 PM)Leffe Wrote: [CITE](#) Did you just post on the wrong thread ? ... ?

Who is talking about CSGO ? What if OP just want to hack a browser-game ?
Who ever wrote ASM on this thread ?

I don't support C# but I can bet there is open-source projects to hook process functions with it.

Don't blame Java, you and I said it : He have to stick to the language of the game.
But we don't even know the target game in the first place !Because I was giving an example man,
rule of thumb if you're going internal (which you should be, really) you stick to language and
preferably compiler of the game you're targetting to minimize compability issues.