Code Interoperability

Mr. S. MOHAN[1], Dr.M. VARATHARAJ[2]
[1]*Assistant professor, Department of computer science*
[2]*Associate professor, Department of EEE, V.S.B. College of Engineering Technical Campus, Coimbatore, Tamil Nadu, India*

***Abstract*—We will discuss code interoperability in one language to another language. Interoperability means managed code with native code. the manage code is managed by CLR in DOT NET (c#). but native code is interoperability which is create by either binary language or c language if it is one language code can connect with another one language code can execute at run time. This code interoperability already designed. If it is needed to link from one language to another language can execute native code. this native code is the library code/binary code. which is managed by CLR/JVM either java or dot net (C#). The native code can connect via through the FORIGN FUNTION INTERFACE (FFI) OR JAVA NATIVE CODE INTERFACE(JNI), both are doing the same function. why need FFI/JNI. now a day these are all most important because of language came to the field which is suitable code can interface and access .so this is the interoperability of code the language either java or python are used interoperability code. Now we will discuss about code interoperability only JAVA. Because it is used to both function FFI/JNI. Why need this two? what is the purpose? what is the benefits?**

***Index Terms*—FFI, JNI CLR, JVM, FFM(API).**

I. INTRODUCTION

In java used native code interface (JNI) that is code interface frame language which is used to native library/binary code link java and execute.it is possible now a day ANDROID written by C/JAVA/KOTLIN, which need can access from native code (binary code). In another one function-foreign function interface. which also interface the language and native code. But entirely differ from the managed code is compile time either CLR (common language run time). JVM (JAVA VIRTUAL MACHINE. This only managed code migrates to another one language, but duffer in JNI/FFM (API) which access the library code can link via java through the JNI/FFM(API).

II. OPERATING SYSTEMS

The operating system which written by either binary or assembly or C languages. That is nature code or binary code. The code executed operating system. The operating system. Means can boot the system or start the system. Which interface between user and system. if any language can execute any one function. That language take use the native code by the language and execute. that process in the function of FFI/JNI

Example: WINDOWS, UNIX, ANDROID.

The android writes by C, JAVA, Kotlin language it not sufficient of code. if need to access to that nature code by that language. because it is the FFM(API)/JNI.

III. FORIGN FUNCTION INTERFACE (FFI)

That mechanism allows to write a program one language to call routing or service are written in another language often used for interacting with dynamic library or native codes.it is interoperability between different programing language by allowing a program to access function and data structure defined another language libraries or code -commonly used c function like python, java or pHp existing c library or dynamic libraries.

IV. JAVA NATIVE INTERFACE (JNI)

The language which allows to java code running with (JVM) to interact with native application and libraries written in other language like c, c++, and assembly. Interface to java to call native method and for native code to call back to java. Loading native code from dynamic shared libraries .it interface access different plat forms interface hardware devices (camera, sensor) other language native drives also.

V. APPLICATION PROGRAMMING INTERFACES (API)

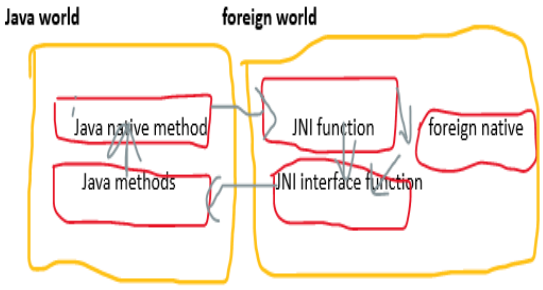
Set of rules and protocols that allows different software application to communicate and interact with each other it is act as intermediation exchange data.one application need to access function and data from anothers.API send send request and other application responds accordingly. a mobile application using and API to come the data from social media platform .it is integration of different software system and services instead of responsibility of existing functionalities. streamless integration and data exchange.

FFM(API)-foreign function and memory application programing interface:

It allows the purpose of interoperability more easily with code and data outside the java run time. FFM-API. Aim to straight forward method for calling native libraries and manipulating the native data. The native function passed run – time provided API to manipulate them. language java only has the JNI built-in. but code changing the dot net family languages can take advantageous FFI.the FFI is by pass all of that. ffi is way off to go and would prepare over the exporting modules specific to the languages.

Forexample

some sdk-software permanent kit to allow to call the c function from c# .it never had to write c# or dot net module just directly call my c function from c#'s.FFI from dylib(dynamic library) and call the right way.



Java cross – language communication frame like as operating system independent or dependent. Because this language easy to use between different languages

The JNI and FFi interface in the JVM standard cross – language frame work between connect one language to another language

VI. IN ANDROID

Built up Linux kernel written primary c and c++ while application frame work and UI (user interface) are primarily with in java increasingly in Kotlin these are written in java add kotlin primary language developed can also c++ performs critical part of this application via through android native development kit. (NDK).

This is another one example for JNI/FFI for the cross- language communication frame work.

VII. STATIC LIBRARY VS DYNAMIC LIBRARY

Static means link directly into executable at compile time. But dynamic library to loaded at the run time. sharing the library cross multiple application.

Linking: static:

Directly incorporated in to the executable during the compilation process many languages copy of the library code separate library t run time library has its own copy it is easy to recompile/re linkinging the entire application example:

.lib(windows)

Linking: dynamic:

Loaded in to memory during the run time are executed only reference to the library function because fill is smaller it is do not contain the library code directly multiple application can sharing same dynamic library memory and disc space updating dynamic library only required updating library file. library can contain to run without re compilation.

Example:

.dll(windows)

.dylib (mac OS)

Dynamic is the context of ffi/jni refer to the ability to load and interact with native library at run time.

Java code to interact with native code written c/c++ and supporting dynamic loading of native library to call function written in a different language and also involve dynamic loading of the libraries.

In jni to load c/c++ library function perform complex operation and call the function from your java code

VIII. CODE INTEROPERABILITY

FFI vs JNI:

Both are doing the same process. but are different in between two that Is the performance and speed of

operation new gen era JNI is OLD one in new already we have to discuss the code interoperability means cross- language frame one line concept.

IX. CONCLUSION

IN PYTHON compiler also have the different language executed with different language code in dot net also have the different code with different language executed in the code. it is the plot form independent/dependent. The operating system one is executed by another one operating system is cross-plot form. But it is like as language is possible cross-language frame work one language executed code if you want to like between another native code can link is possible to execute. That process is told FFI (foreign function interface) or JNI (java native code interface. native code either c or assembly language which link by java or other language. In ANDROID written by c, c++ and java increased by KOTLIN languages, because the native code link by this process. In new generation edition will use FFI. Because java has the JNI.the compare to this FFI is function the JNI.

REFERENCES

[1] Dante Broggi, Possible violation of the pthread_create interface? #4324, 2023, [online] Available: https://github.com/ponylang/ponyc/issues/4324.
[2] Pony Developers. Reference capabilities: The list of reference capabilities, 2023, [online] Available: https://tutorial.ponylang.io/reference-capabilities/reference-capabilities.html\#the-list-of-reference-capabilities.
[3] Alex Gaynor, what is memory safety and why does it matter? [online] Available: https://www.memorysafety.org/docs/memory-safety.
[4] Zach Marzouk, move away from memory-unsafe languages like c and c++ nsa urges, 2021, [online] Available: https://www.itpro.com/development/programming-languages/369499/move-away-from-memory-unsafe-languages-c.
[5] Dan Lorenc, Trends challenges and shifts in the software vulnerability mitigation landscape, 2021, [online] Available: https://github.com/Microsoft/MSRC-Security-Research/blob/master/presentations/2019_02_BlueHatIL/2019_01\%20-\%20BlueHatIL\%20-\%20Trends\%2C\%20challenge\%2C\%20and\%20shifts\%20in\%20software\%20vulnerability\%20mitigation.pdf.
[6] Pony Developers. Reference capabilities: The list of reference capabilities, 2023, [online] Available: https://tutorial.ponylang.io/reference-capabilities/reference-capabilities.html\#the-list-of-reference-capabilities.