TEST

by Ian Cee

Submission date: 25-Jun-2019 12:09PM (UTC+0300)

Submission ID: 1146920041

File name: ABSTRACT.docx (184.31K)

Word count: 1909

Character count: 10334

TOWARDS PLATFORM INDEPENDENT PROGRAMMING

Frankline O. Momanyi

Department of Computer Science and Statistics, Moi University, Eldoret, Kenya

ABSTRACT

The ability of a programming language to execute across multiple platforms with minimum or no modification is what is called Platform independence in Information Technology. This portability of a language is very important as it does not restrict users to specific environments. This feature also prevents glitches and errors when users run these programs in various environments. There are very many operating systems in the market today meaning it Is practically impossible to develop applications for every operating system that exists in the market. Building one software that runs in all environments is much easier that building multiple software's that perform the same functions that run in all these operating systems. It is also fundamental to developers who want to reach big audiences to use their applications as developers don't really know exactly what environments their target audiences run on. An application will get more users when it is platform independent as it will literally run anywhere unlike applications that are platform dependent which will restrict the users to environments.

KEYWORDS

INTRODUCTION

This paper does justice to the mentioned topic in two ways: looking at Java as a platform independent language and discussing platform independence from a server scripting perspective. The whole point of platform independence heavily relies on how a programming language is compiled. For instance, the C and the C++ programming languages are platform independent on the code level but after compilation they become very platform dependent. When a developer compiles a C or a C++ program, it is immediately converted to a machine-readable language that has a .exe extension which is very specific to the windows platform thus cannot run on a Linux environment.

Java on the other hand is the most famous platform independent language because it can be executed anywhere regardless of the environments. The platform independence property lies solely on the compiler that compiles Java. The java compiler compiles the java source code into bytecode. It is this bytecode that runs in the Java Runtime Environment whose most important component is the Java Virtual Machine (JVM) which is used to analyze and execute Java Byte Code. The Javac compiler saves the bytecode in the local computers disk with the extension .class (which is converted from the .java extension). When the program is to be run, the bytecode is converted using the Just in Time (JIT) compiler. The bytecode is not normally in an executable stage.

The main purpose of generating byte code for a program compiled is to achieve platform independency that means this byte code generated in one platform can be executed in another. The one which makes the byte code generated in Windows OS to be executed in the UNIX OS is the JVM of UNIX platform. From this statement, you may have understood that JVM is platform dependent and the byte code generated by Java program is platform independent. The Byte code generated can run on any JVM irrespective of to which platform the JVM belongs. Whatever the JVM in which the byte code runs the output remains same.

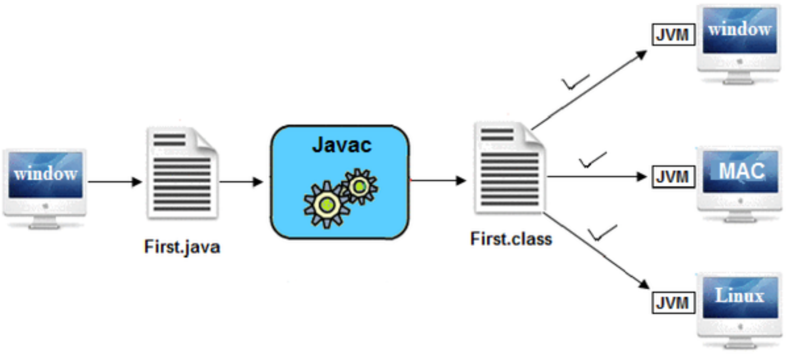


Figure 1 https://www.tutoropedia.com/platform-independent

FINDINGS

JAVA AS A PLATFORM INDEPENDENT LANGUAGE

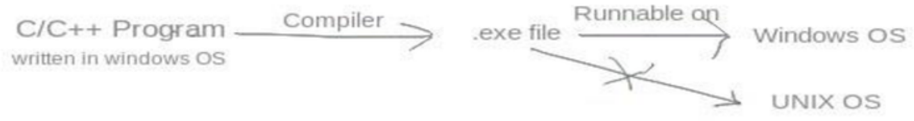
Java has been called as the platform independent machine because of the Java Virtual Machine (JVM). Every developer who uses Java knows that Java bytecode runs in a Java Runtime Environment (JRE). The most important element of the Java Runtime Environment is Java Virtual Machine (JVM), which analyzes and executes Java byte code. That byte code can be run on any platform.

Java source code is compiled into bytecode when we use the javac compiler. The bytecode gets saved on the disk with the file extension .class. When the program is to be run, the bytecode is converted, using the just-in-time (JIT) compiler. The result is machine code which is then fed to the memory and is executed. Before understanding how Java is platform independent, you first need to know,

1. What is platform?
2. How C/C++ program is executed?
3. Difference between byte code and native code?
4. How Java program is executed?

A platform is combination of processor and OS (operating system). In general, it can be said that it is the hardware or software component in which programs run.

When you write program in C/C++ and when you compile it, it is directly converted into machine readable language(.exe). This .exe file generated is specific to the operating system like for example, when you compile program in windows OS, the .exe file generated for that program is specific to only windows OS and cannot be made to run in UNIX OS.

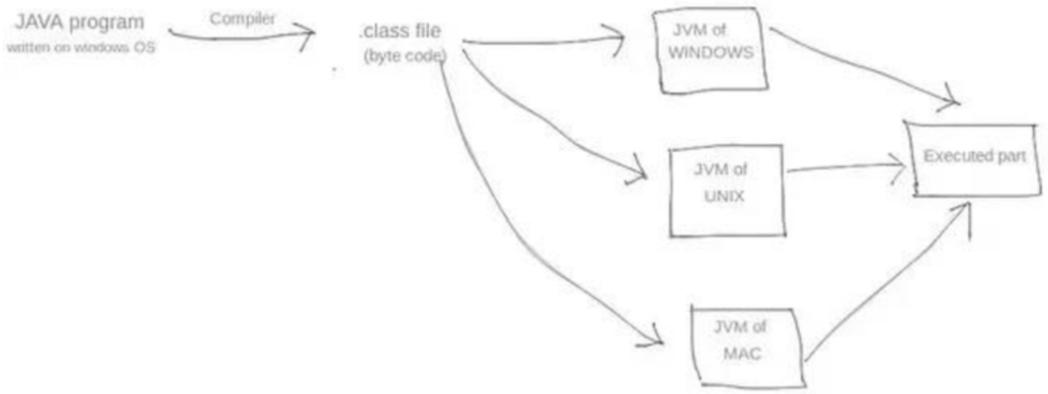


That’s why C/C++ programs are platform dependent. Here .exe file is the Native code.

Native code is similar to machine code i.e codes that is understood by machine. Native codes are specific to platform in that, a native code generated by a program for Windows Operating System is different from Native code generated for the same program for Unix OS.

Byte codes are intermediate codes generated after compilation and it is not the executable code like Native code. The Byte code requires a virtual machine to execute in machine. Byte codes generated by one platform can be executed in another platform also.

When you write program in JAVA and when you compile it, a separate file is created for the program compiled, this file(.class) is known as byte code of java. This byte code will not be in executable stage. The main purpose of generating byte code for a program compiled is to achieve platform independency that means this byte code generated in one platform can be executed in another. The one which makes the byte code generated in Windows Operating System to be executed in the UNIX Operating System is the Java Virtua Machine of UNIX platform. From this statement, it is understood that JVM is platform dependent and the byte code generated by Java program is platform independent. The Byte code generated can run on any Java Virtual Machine irrespective of to which platform the JVM belongs. Whatever the JVM in which the byte code runs the output remains same.



SERVER-SIDE SCRIPTING AS A PLATFORM INDEPENDENT PROGRAMMING LANGUAGES

When you type in a URL, lots of code is at work to bring a page to your screen. What connects your site's database to the browser, creating a smooth, user-friendly experience? That's the software built by **server-side scripts,** languages that build your site behind the scenes. The goal of this software? To provide a seamless experience for the user that's as close to a desktop application as possible.

There are many server-side languages working toward that end goal. The language you choose for your site depends on a mix of your site's requirements, your database/operating system setup, and the preferences of your development team. Knowing what each script can offer and what sets it apart is helpful in deciding how to build your back end, and who to hire.

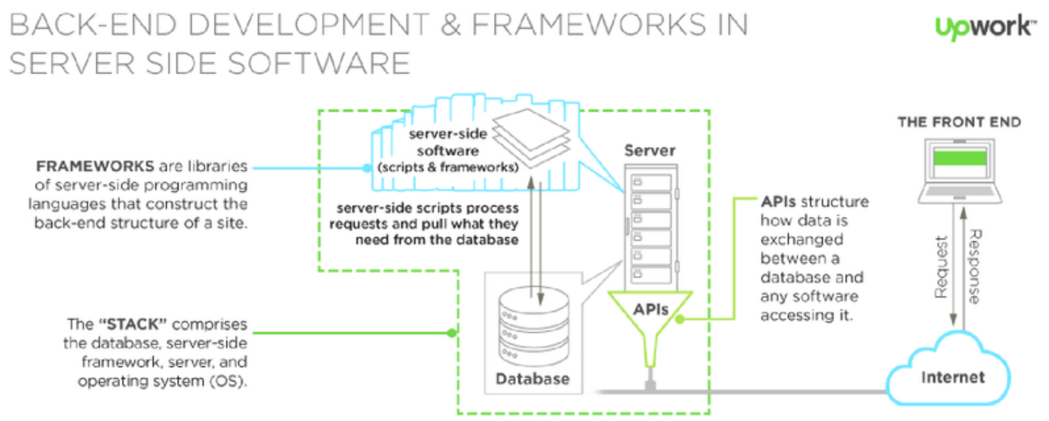

BACK-END DEVELOPMENT & FRAMEWORKS IN SERVER SIDE SOFTWARE
upwork
FRAMEWORKS are libraries of server-side programming languages that construct the back-end structure of a site.
server-side software (scripts & frameworks)
server-side scripts process requests and pull what they need from the database
Server
THE FRONT END
APIs structure how data is exchanged between a database and any software accessing it.
Request
Response
APIs
The "STACK" comprises the database, server-side framework, server, and operating system (OS).
Database
Internet

Figure 2 https://www.upwork.com/hiring/development/server-side-scripting-back-end-web-development-technology/

SERVER-SIDE CODE AND DATABASES

How it works: If a database is a site’s library, server-side scripting processes what the user is looking for via the server, then locates the book, chapter, page, and exact line of data, delivering that information back to the browser. It’s designed to be smooth, fast, and seamless.

This is all possible through the server-side software and middleware your back-end developer writes, which create a tailored channel from site to database. Information for your site resides on the server until it’s requested, which makes your site both fast and secure.

Popular server-side languages

PHP: The most popular server-side language on the web, PHP is designed to pull and edit information in the database. It’s most commonly bundled with databases written in the SQL language. PHP was designed strictly for the web and remains one of the most widely used languages around. It’s easy to install and deploy, is staying competitive with lots of modern frameworks, and is the foundation for a number of content-management systems. PHP-powered sites: WordPress, Wikipedia, Facebook

Python: With fewer lines of code, the Python programming language is fast, making it ideal for getting things to market quickly. The emphasis is on readability and simplicity, which makes it great for beginners. It’s the oldest of the scripting languages, is powerful, and works well in object-oriented designs. Python-powered sites: YouTube, Google, The Washington Post

Ruby: If you’re expecting complicated logic on the database side of your site, the Ruby programming language is an excellent option. Unlike Python, Ruby is equal parts simplicity and complexity, pairing simple code with more flexibility and extra tools. Ruby bundles the back end with database functionality that PHP and SQL can offer as a pair—it’s great for startups, easy maintenance, and high-traffic demands. It requires developers to use the Ruby on Rails framework, which has vast libraries of code to streamline back-end development. Ruby-powered sites: Hulu, Twitter (originally), Living Social, Basecamp

C#: The language of Microsoft's .NET Framework—the most popular framework on the web—C# combines productivity and versatility by blending the best aspects of the C and C++ languages. It's excellent for developing Windows applications, and can be used to build iOS, Android mobile apps with the help of a cross-platform technology like Xamarin.

C++: Great for complex applications also built on the .NET Framework, the C++ programming language is a difficult but high-powered language that works well for data-heavy sites. Speed is central to C++, and it runs well alongside other languages like Java and Python.

Java: A subset of the C language, Java comes with a huge ecosystem of add-on software components. At its core, Java is a variation of C++ with an easier learning curve, plus, it's platform independent thanks to the Java Virtual Machine. "Compile once, run anywhere" is its motto—and it's excellent for enterprise-level applications, high-traffic sites, and Android apps. Java sites: Twitter, Verizon, AT&T, Salesforce

Erlang: A general-purpose programming language, Erlang is also a concurrent language, which means several processes can run simultaneously on the language-level without external library support. It's used in the LYME and LYCE stacks, numerous CMS and databases, GitHub, Facebook chat, and Goldman Sachs, supporting its high-frequency trading requirements.

SERVER-SIDE FRAMEWORKS

Ruby on Rails: This Ruby framework is the overriding way to implement Ruby. Its "gems" include plug-ins and libraries of code that streamline development.

ASP.NET: This Microsoft framework is the most popular enterprise-level framework—it supports multiple programming languages simultaneously for one project. So, the same application can be built with both C# and C++, via CLI (common language interface). It's most recent iteration, ASP.NET 5, is now open to non-Windows platforms for the first time.

Django: This Python framework was developed to meet the needs of development in a fast-paced environment. Django sites: Pinterest, Nasa, Pitchfork

Node.js: JavaScript is typically a front-end script, but with the Node.js framework, it can be used in server-side technology, from APIs to entire stacks. Its core selling point is how it handles clientserver communication—it’s fast, doesn’t bottleneck, and is ideal for real-time apps like chat rooms, data-heavy applications, and any software that requires the streaming of fresh content, like a news feed. Node.js sites: Dow Jones, PayPal, LinkedIn

Express.js & Koa: These JavaScript-powered middleware frameworks work on top of the Node.js development environment and control the flow of information on the back end of a site.

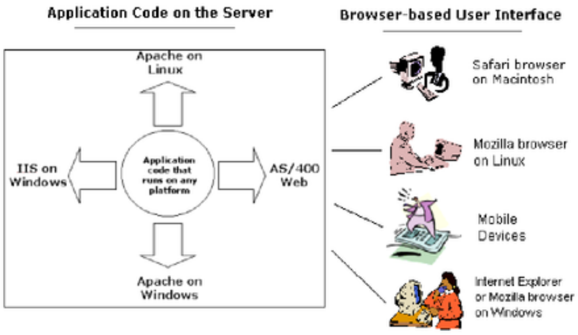

Application Code on the Server
Browser-based User Interface
Apache on Linux
IIS on Windows
Application code that runs on any platform
AS/400 Web
Apache on Windows
Safari browser on Macintosh
Mozilla browser on Linux
Mobile Devices
Internet Explorer or Mozilla browser on Windows

Figure 3 Figure 2 https://www.tutoropedia.com/platform-independent

Content from https://www.upwork.com/hiring/development/server-side-scripting-back-end-web-development-technology/

CONCLUSIONS AND FUTURE WORK

REFERENCES

Tutoropedia. (2019). Platform Independent Programming. Retrieved June 24, 2019, from https://www.tutoropedia.com/platform-independent

TEST

ORIGINALITY REPORT

55%	53%	0%	54%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.answerspoint.com Internet Source	39%
2	www.upwork.com Internet Source	7%
3	Submitted to Info Myanmar College Student Paper	4%
4	en.wikibooks.org Internet Source	2%
5	Submitted to Career Education Corporation Online Student Paper	1%
6	Submitted to Herefordshire College Of Technology Student Paper	1%
7	stackoverflow.com Internet Source	<1%
8	Submitted to Blackpool and The Fylde College, Lancashire Student Paper	<1%

9	Submitted to Coventry University Student Paper	<1%
10	Submitted to Durban University of Technology Student Paper	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

TEST

GRADEMARK REPORT

FINAL GRADE

/5

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10