

Working of Java Virtual Machine (JVM) & its Architecture

In order to write and execute a software program you need the following

- 1) **Editor** – To type your program into, a notepad could be used for this
- 2) **Compiler** – To convert your high language program into native machine code
- 3) **Linker** – To combine different program files reference in your main program together.
- 4) **Loader** – To load the files from your secondary storage device like Hard Disk, Flash Drive, CD into RAM for execution. The loading is automatically done when you execute your code.
- 5) **Execution** – Actual execution of the code which is handled by your OS & processor.

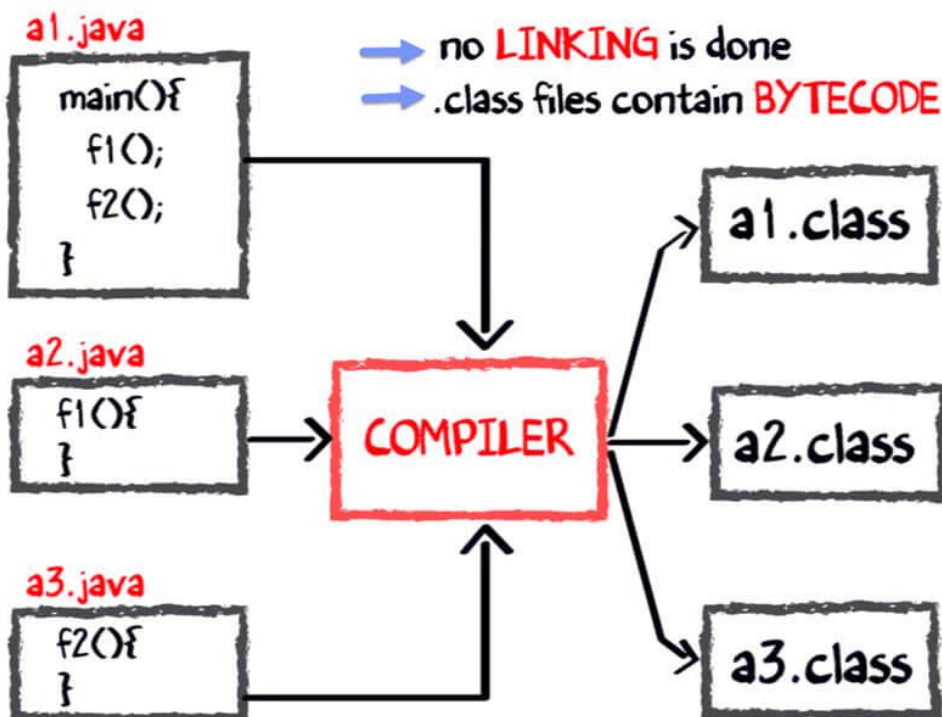
Java code compilation and execution in Java VM (JVM)

Let's look at the process for JAVA. In your main you have two methods f1 and f2.

main method is stored in file a1.java

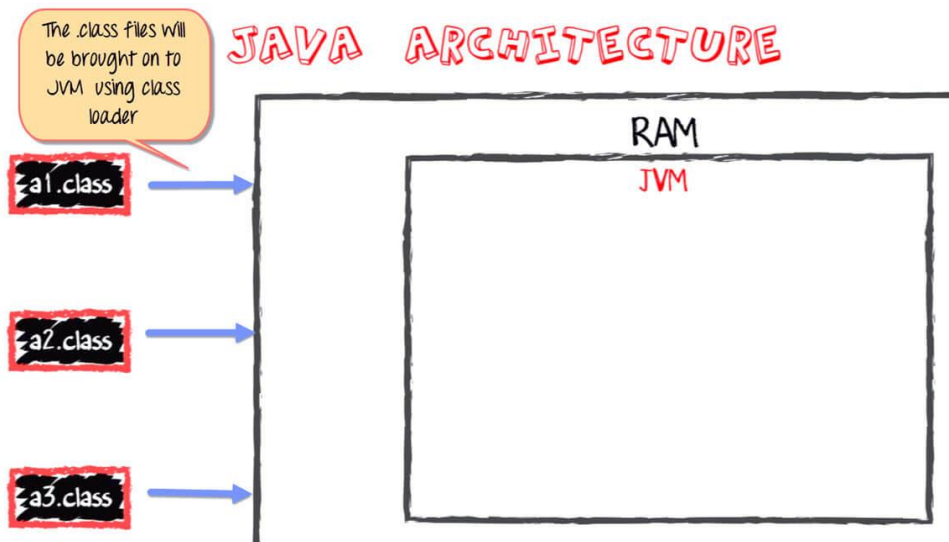
f1 is stored in file as a2.java

f2 is stored in file as a3.java



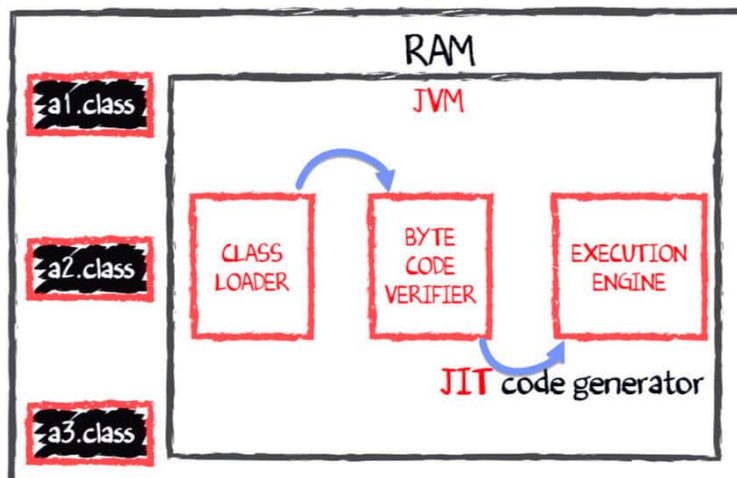
The compiler will compile the three files and produces a corresponding .class file which consists of BYTE code. Unlike C, no linking is done.

The Java VM or Java Virtual Machine resides on the RAM. During execution, using the class loader the class files are brought on the RAM. The BYTE code is verified for any security breaches.



Next, the execution engine will convert the Bytecode into Native machine code. This is just in time compiling. It is one of the main reason why Java is comparatively slow.

JIT converts **BYTECODE** into machine code



NOTE: JIT or Just-in-time compiler is the part of the Java Virtual Machine (JVM). It is used to speed up the execution time. It interprets part of the Byte Code that has similar functionality at the same time.

What is JVM?

JVM stands for **J**ava **V**irtual **M**achine. It is the engine that drives the Java Code. It converts Java bytecode into machines language.

- ❖ In other programming language, the compiler produces code for a particular system. But Java compiler produces code for a Virtual Machine.
- ❖ In JVM, Java code is compiled into bytecode. This bytecode gets interpreted on different machines
- ❖ Between host system and Java source, Bytecode is an intermediary language.
- ❖ JVM is responsible for allocating a memory space.



Working of Java Virtual Machine(JVM) & its Architecture

Why is Java both interpreted and compiled language?

Programming languages are classifies as :

- Higher Level Language - C++ , Java
- Middle Level Languages - C
- Low Level Language - Assembly
- Finally the lowest level as the Machine Language.

A compiler is a program which converts a program from one level of language to another. *example* conversion of C++ program into machine code.

The java compiler is a convert's high level java code into bytecode (which is also a type of machine code).

A interpreter is a program which converts a program at one level to another programming language at the same level. *example* conversion of Java program into C++

In Java , the Just In Time Code generator converts the bytecode into the native machine code which are at the same programming levels.

Hence java is both compiled as well as interpreted language.