ABOUT JVM AND ITS MEMORY USAGE

JVM 과 JVM 의 메모리 관리에 관하여

Nathan Cho *조나단 [20425]*
Department of Software
Sunrin Internet High School
Seoul, Korea
dev.bedrock@gmail.com

목차

1. JVM

1) JVM 이란

Java Virtual Machine (JVM)은 자바 프로그램을 실행 할 수 있는 추상적 컴퓨터이다.

쉽게 설명하자면 JVM 은 Java 플랫폼에 없어서는 안될 기초이다. '추상적 컴퓨터'라는 말을 설명해 보자면 일반적인 컴퓨터처럼 JVM 은 스스로의 명령어 세트를 가지고 있고 실행하면서 메모리를 할당한다. 즉 작동하는 장치 안에 또 다른 컴퓨터를 가지게 해주는 것이다.

JVM 은 프로그래머가 작성한 *.java* 코드를 컴파일 된 *.class* 파일을 실행하고 결과를 출력하는 기계 안의 또 다른 기계이다.

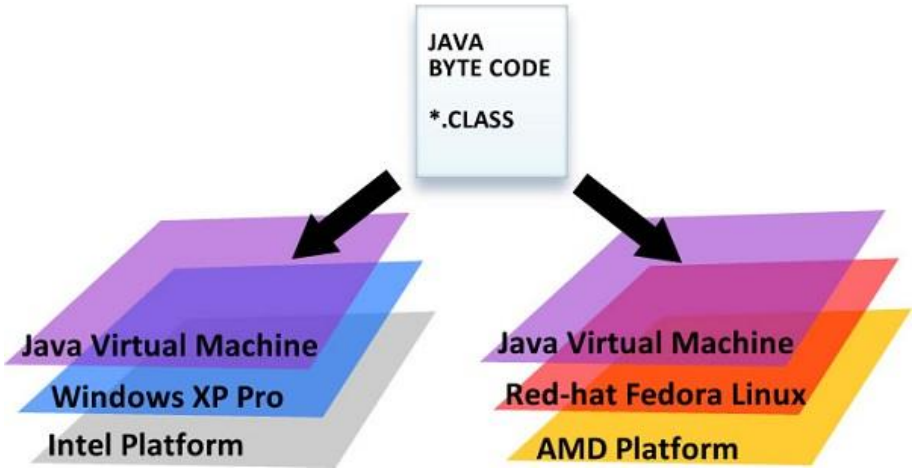

그림 1-1. CPU 가 다르고 운영 체제가 다르더라도 JVM 환경에서 같은 코드를 실행 할 수 있다.
(출처: https://thecustomizewindows.com/2012/08/java-virtual-machine-overview-and-principle/)

2) JVM 의 특성

JVM 은 최초의 가상 머신이 아닌 다른 여러 가지 가상 머신에게서 영향을 받고 만들어 졌다. JVM 아키텍처 설계자 *James Gosling* 은 기존에 존재하던 *USCD p-code* 와 *Smalltalk VM* 에서 많은 영향을 받았다. 이에 따라서 JVM 은 이 두 개의 가상 머신의 단점들을 보완하고 추가된 기능을 가지고 있다.

간단한 주요 특성은 다음과 같다.

- 스택 기반 가상 머신
- 객체 지향 프로그래밍을 구현

- 제한적인 포인터 사용
- 가비지 컬렉션
- 철저한 타입 정의로 플랫폼 독립적
- 독자적인 *Bytecode verifier (바이트코드 검증기)*로 안정적

JVM 은 하나의 규격에서 모든 것이 실행되는 것이 아닌 몇 가지의 규격이 있다. 주요적인 규격으로는 가장 많이 사용되는 Java SE, Java EE 등이 있고 사용 목적에 따라서 분류된다.

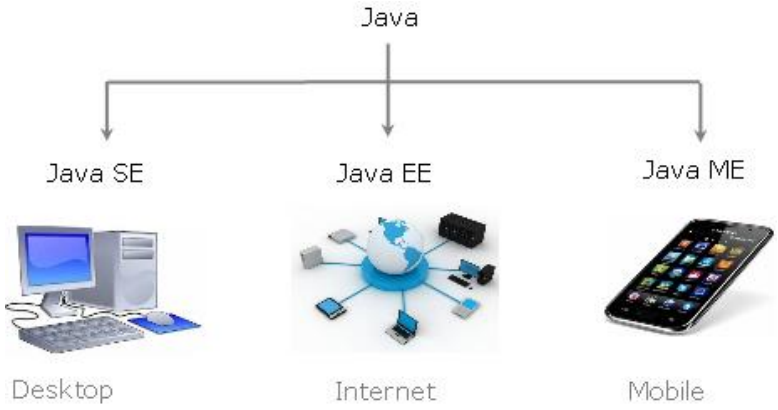

그림 1-2. 여러 가지 JVM 버전들 (출처: http://codingfox.com/0-5-hisory-of-java/)

또한 Java 언어 하나만 지원하는 것이 아닌 많은 스크립트 언어와 Java 를 기반으로 한 언어들을 지원한다. *(Groovy, Scala, Kotlin, 등)*

3) JVM 의 구조

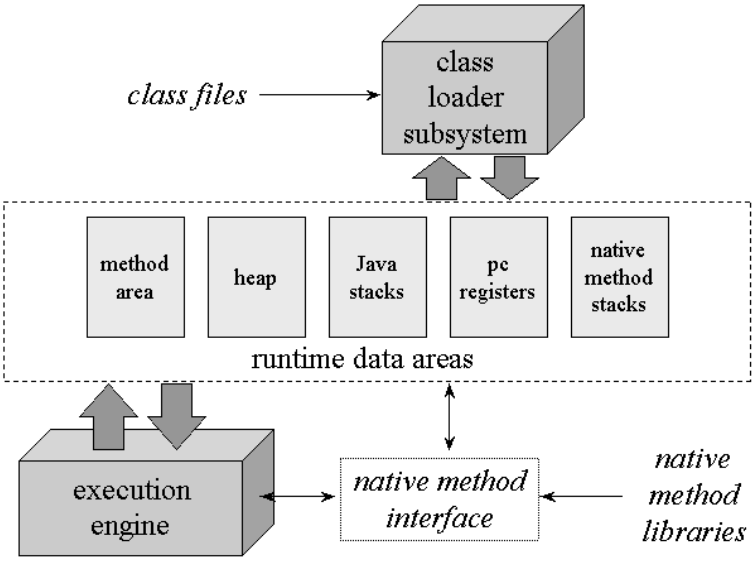

그림 1-3. 클래스 파일을 실행하는 식으로 동작하는 JVM 아키텍처 (출처: http://www.artima.com/insidejvm/ed2/jvm2.html)

클래스 데이터를 메모리로 로딩해서 실행하는 단순하고도 복잡한 구조이다. 기본적인 컴퓨터에서 볼 수 있는 구조를 JVM 은 '가상 머신'이라는 기술로 컴퓨터 위의 컴퓨터에서 코드를 실행한다.

2. JVM 의 메모리 관리

JVM 의 큰 특징 중 하나인 *Garbage Collector (가비지 콜렉션)* 은 단순히 사용되지 않는 자원을 자동으로 반환하는 기능이다. 쉽게 말하면 자동 메모리 관리인 셈이다.

가비지 콜렉션이 작동하는 원리는 단순하다. Java 프로그램이 실행되는 중에 코드에서 객체를 생성하고 다시 사용되지 않는다면 JVM 은 그 객체를 "쓰레기"로 분류한다. 코드가 실행되는 중에 하드웨어 자원이 충분하지 못해 더 이상 객체를 생성 할 수 없을 때 가비지 콜렉션이 작동한다. 가비지 콜렉터는 "쓰레기"로 분류된 객체들을 찾아서 메모리에서 반환시키고 다른 객체를 생성할 메모리상의 공간을 만든다.

JVM 을 사용하지 않는 다른 언어를 예로 들자면 *C* 의 *malloc* 와 *free* 함수가 있다. 프로그래머가 프로그램이 실행 중일 때 메모리를 사용하는 것까지 생각해서 프로그램을

작성해야 한다. 하지만 JVM 을 사용한다면 JVM 의 가비지 콜렉션 기능이 메모리 관리를 스스로 해 주어 프로그래머들이 편하다는 장점이 있다.

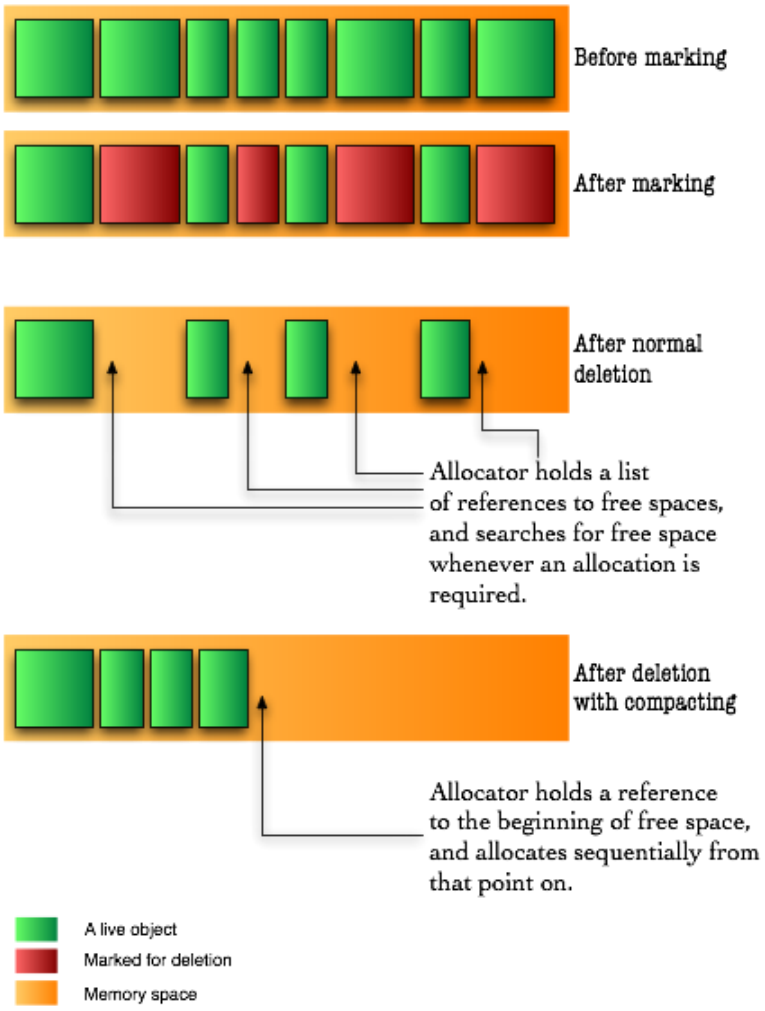

그림 2-1. 초록색이 생성된 객체, 빨간색이 "쓰레기"로 분류된 객체이다. 가비지 콜렉션이 수행되고 나서 빨간색 "쓰레기"로 분류되었던 객체들이 메모리에서 반환된 것을 관찰 할 수 있다. (출처: https://stackoverflow.com/questions/4813005/garbage-collection-java)