Виртуальная машина Java: Основы

Описание

Понимание виртуальной машины Java (JVM) – это именно то, на что она похожа: машина, которая виртуально находится внутри программного обеспечения. Подобно сознанию Нео в Матрице, это цифровая машина, существующая внутри более крупной операционной системы. Это программное обеспечение устанавливается на локальном устройстве и может запускать другие операционные системы в рамках существующей. Конкретным и распространенным случаем использования является установка VirtualBox на Windows или Mac, а затем создание виртуального экземпляра другой операционной системы, например, дистрибутива Linux. Если вы здесь, то это потому, что вы выбрали красную таблетку и хотите исследовать кроличью нору Страны чудес Java. В этой заметке мы рассмотрим виртуальную машину Java, как она работает и как устранять ошибки виртуальной машины Java.

Что такое виртуальная машина Java?

Виртуальная машина Java (JVM) – это программа, которая интерпретирует байткод Java для запуска в качестве программы, предоставляя среду выполнения, которая выполняет этот процесс. Более того, она отделена от операционной среды, поддерживая философию “напиши один раз, запусти где угодно”. Одним из наиболее значительных преимуществ использования Java является использование JVM для запуска Java-программы в любой операционной среде. В следующих разделах мы обсудим, для чего используется JVM, и основы ее понимания.

Виртуальная машина Java – это вирусный и широко используемый инструмент, поддерживаемый разработчиками из корпорации Java и сообществом разработчиков с открытым исходным кодом. Компания Sun Microsystems решила сделать Java открытым исходным кодом, что привело к разработке OpenJDK. Продолжающееся использование и поддержка OpenJDK в основном обусловлены тем, что основная нагрузка ложится на разработчиков Oracle.

Для чего используется JVM?

Виртуальные машины Java служат двум основным целям; первая – предоставить средства для выполнения Java-программы в любой среде. Вторая – поддержание и оптимизация памяти программы. Во времена зарождения Java философия “пиши один раз, выполняй везде” стала революционным изменением, которое изменило ландшафт разработки программ. Программы, написанные до появления этой новой философии, работали только на определенных целевых платформах. Вместо этого разработчикам приходилось управлять памятью программы, что отнимало много времени.

Это изменение означает, что у разработчиков стало на одну сложную вещь меньше, которую нужно учитывать при кодировании. Кроме того, новый подход поддерживает принцип разработчиков “напиши один раз, запусти везде”, что устраняет необходимость переписывать код для уникальных операционных сред. Обычно считается, что JVM имеет двойное определение – техническое и неформальное, призванное осветить ее использование в зависимости от пользователя и его поведения.

- **Техническое определение JVM:** JVM – это спецификация программы, которая обеспечивает среду выполнения для выполнения кода Java.
- **Неофициальное определение JVM:** JVM выполняет программы Java, используя настроенные параметры для управления ресурсами программы во время выполнения.

Очень часто принято думать и говорить о JVM как о процессе, запущенном на компьютере или сервере для управления использованием ресурсов Java-приложения. Спецификация JVM описывает требования, необходимые для создания программы, выполняющей эти задачи.

Память и сборка мусора

Наиболее распространенным взаимодействием с JVM является наблюдение за использованием памяти в “куче и стеке” и настройка параметров памяти JVM. Память JVM управляется с помощью сборки мусора, что отличается от методов, использовавшихся в предыдущих языках. В прошлом память программы была задачей, оставленной разработчикам. Сборка мусора – это процесс, который постоянно следит за программой на предмет неиспользуемой памяти и удаляет ее для повышения производительности. Процесс сборки мусора выполняется внутри JVM, а не в программе. Этот процесс сборки мусора использует спецификации разработчиков и операторов для индивидуального использования программы.

Java не “близка к металлу”

Предыдущие языки, такие как C, C++ и другие, считаются “близкими к металлу”, что означает, что они работают намного быстрее. Кроме того, эти языки могут напрямую управлять памятью программы, а код Java – нет. Java оставляет эту возможность JVM, что было концепцией, которая подверглась обстрелу во время ее создания, поскольку она ограничивала контроль программистов над управлением памятью.

С тех пор Java добилась значительного прогресса в улучшении процесса сборки мусора и управления памятью. Благодаря постоянной поддержке и развитию процесс был значительно усовершенствован и продолжает совершенствоваться, выравнивая ландшафт. Виртуальная машина Java использует так называемый компилятор Just-In-Time, который компилирует байткод в машинный код для операционной среды. Это делается для увеличения скорости выполнения кода, работающего в JVM.

Архитектура виртуальной машины Java

Понимание виртуальной машины Java становится проще, если понять ее архитектуру и то, как она функционирует. В оставшейся части этой заметки мы обсудим, как работает JVM и как архитектура влияет на выполнение Java-программ.

Загрузчик классов

Загрузчик классов используется для загрузки файлов классов. Файлы классов необходимы загрузчику классов для выполнения трех основных функций – связывания, загрузки и инициализации.

Область методов

Область методов JVM – это место, где находятся структуры классов различных типов, необходимые для выполнения java-программы.

Куча

Все объекты, связанные с ними переменные экземпляра и массивы хранятся как общая память в куче, где она разделяется между несколькими потоками. Потоки создаются для разделения различных задач памяти и, таким образом, хранятся отдельно за пределами общей памяти.

Языковые стеки JVM

Языковые стеки Java хранят локальные переменные и их частичные результаты. Каждый поток имеет свой собственный стек JVM, создаваемый по мере создания потока. Когда начинается вызов метода, создается новый фрейм, который затем удаляется по завершении вызова метода.

Регистры ПК

Регистр PC хранит адрес текущей исполняемой инструкции виртуальной машины Java. В Java каждый поток получает свой собственный регистр PC.

Стеки нативных методов

Стеки нативных методов хранят инструкции нативного кода, написанного не на Java, а на другом языке, с использованием нативной библиотеки.

Механизм исполнения

Execution Engine – это тип программного обеспечения, используемый для тестирования аппаратных средств, программного обеспечения или целых систем; он делает это, не сохраняя никакой информации о тестируемом продукте.

Интерфейс нативных методов

Интерфейс нативных методов – это основа программирования, которая позволяет выполнять код Java в JVM для вызова библиотек и нативных приложений.

Библиотеки нативных методов

Native Libraries – это коллекция нативных библиотек, таких как языки C, необходимых механизму исполнения.

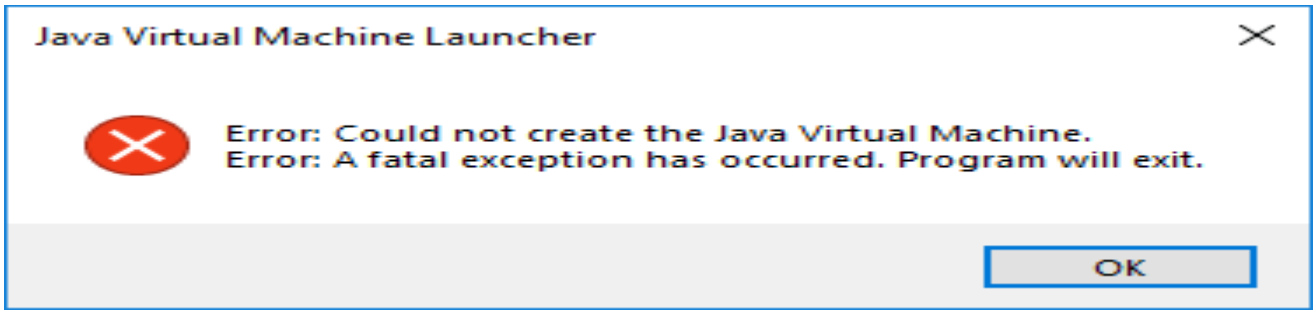
Ошибки виртуальной машины Java

Ошибка виртуальной машины Java, также известная как ошибка JVM, классифицируется как ошибка, генерируемая виртуальной машиной Java. Когда возникает ошибка такого типа, это обычно означает, что компьютер не может прочитать или понять код.

Ошибка запуска виртуальной машины Java

Ошибка запуска Java-машины возникает из-за того, что необходимые компоненты недоступны при запуске JVM. Например, если при запуске отсутствуют какие-либо необходимые классы, JVM выдаст ошибку запуска, чтобы предупредить пользователя или оператора о проблеме. Это происходит до завершения запуска и обычно является результатом того, что запуск не может быть завершен.

Вот пример того, как выглядит ошибка запуска JVM.



Выводы по виртуальной машине Java

В заключение давайте рассмотрим наиболее важные моменты, которые следует вынести из этой статьи. Мы много говорили о JVM, но есть несколько ключевых моментов, которые должны выделяться на фоне остальных, поскольку они являются основой для понимания этих концепций.

- Полная форма JVM – Java Virtual Machine. JVM – это механизм, который преобразует байткод Java в машинный язык.
- Архитектура JVM в Java содержит загрузчик классов, область методов, кучу, стеки языка JVM, регистры ПК, стеки нативных методов, механизм выполнения, интерфейс нативных методов, библиотеки нативных методов.
- В JVM код Java компилируется в байткод, который может быть интерпретирован на разных машинах.
- JIT расшифровывается как Just-in-time compiler и является частью виртуальной машины Java (JVM), она используется для ускорения времени выполнения.
- По сравнению с другими компилирующими машинами, JVM в Java может иметь более низкую скорость выполнения.

Дата Создания
19.04.2023