TOP 20 INTERVIEW QUESTIONS ON JVM ARCHITECTURE

YOUTUBE VIDEO - https://www.youtube.com/watch?v=Ms7KDg_TVcI

What is JVM, JDK, and JRE?

- **JVM**: Java Virtual Machine that runs Java bytecode.
- **JDK**: Java Development Kit, includes tools like compilers and debuggers.
- **JRE**: Java Runtime Environment, provides libraries and JVM to run Java applications.

Explain the architecture of JVM.

- JVM consists of:
    - **ClassLoader**: Loads class files.
    - **Memory Area**: Method area, heap, stack, program counter, native method stack.
    - **Execution Engine**: Interprets or compiles bytecode into machine code.
    - **Garbage Collector**: Manages memory by removing unused objects.

What are the components of the JVM memory model?

- **Heap**: Stores objects and class-level variables.
- **Method Area**: Stores class-level data like runtime constant pool, field, method data, and constructor code.
- **Stack**: Stores local variables, method call information.
- **Program Counter (PC) Register**: Holds the address of the JVM instruction being executed.
- **Native Method Stack**: Stores native method information.

What is the role of ClassLoader in JVM?

- It loads `.class` files into JVM, breaking it into three parts: **Bootstrap ClassLoader**, **Extension ClassLoader**, and **Application ClassLoader**.

Explain the different memory areas in JVM.

- **Heap**: Object memory.
- **Stack**: Thread-specific memory for method invocations.
- **Method Area**: Stores class-related data.
- **PC Register**: Keeps track of JVM instruction addresses.
- **Native Method Stack**: Used for native (non-Java) code.

What is the difference between Stack and Heap memory?

- **Stack**: Stores local variables and method calls; operates on LIFO.

- **Heap**: Stores objects; shared among all threads.

How does JVM handle method invocation?

- JVM uses the **Stack** memory for method invocation, creating a new stack frame for every method call.

What is the Execution Engine in JVM?

- Converts bytecode to machine-specific code, consisting of:
    - **Interpreter**: Executes bytecode line by line.
    - **JIT Compiler**: Converts bytecode into native machine code for high performance.
    - **Garbage Collector**: Manages memory deallocation.

What is JIT (Just-In-Time) Compilation?

- Part of the execution engine, **JIT** compiles frequently used bytecode to machine code at runtime for improved performance.

How does JVM manage memory?

- JVM uses **Garbage Collection** to automatically free memory by identifying and deleting unused objects in the heap.

What are the different types of ClassLoaders?

- **Bootstrap ClassLoader**: Loads core Java classes (`java.lang.*`).
- **Extension ClassLoader**: Loads classes from the Java extension libraries.
- **Application ClassLoader**: Loads application-specific classes from the classpath.

What is the role of the Garbage Collector in JVM?

- It automatically reclaims memory by removing objects that are no longer in use, ensuring efficient memory management in the heap.

What are the phases of Garbage Collection in JVM?

- **Mark**: Identifies objects that are in use.
- **Sweep**: Removes objects that are not marked.
- **Compact**: Reorganizes memory by moving active objects together.

Can you manually trigger Garbage Collection in Java?

- You can request garbage collection using `System.gc()`, but there is no guarantee that it will run immediately.

What is the PermGen (Permanent Generation) space in JVM?

- **PermGen** is a non-heap memory area that stores metadata about classes and methods. It was removed in Java 8 and replaced by **Metaspace**.

What is the difference between PermGen and Metaspace?

- **PermGen** was fixed in size and prone to `OutOfMemoryError`. **Metaspace** grows dynamically, improving memory management.

What is the role of the Program Counter (PC) Register in JVM?

- PC Register holds the address of the next instruction that the JVM will execute, maintaining the execution flow for each thread.

What happens during JVM startup?

- The JVM:
    - Loads the main class.
    - Uses the ClassLoader to load classes.
    - Initializes class and instance variables.
    - Executes the `main()` method of the class.

What are the different types of Garbage Collectors in JVM?

- **Serial GC**: Suitable for single-threaded applications.
- **Parallel GC**: Uses multiple threads for garbage collection.
- **CMS GC (Concurrent Mark Sweep)**: Minimizes pauses by doing most work concurrently.
- **G1 GC (Garbage First)**: Default collector in Java 9+, divides the heap into regions for better performance.

How does the JVM handle multithreading?

- JVM creates separate stacks for each thread, and manages synchronization through monitors and locks to ensure thread safety.