* System Proggramming and Operating System (SPOS) - Case Study - 3

Name :- Kaustubh Shrikant Kabra.
Class :- Third Year Engineering.
Div :- A Roll Number :- 38
Batch :- T2
Department :- Computer Department :
College :- AISSMS's IOIT.

Study the concepts of class loading in Java

Java Class Loader -

Java class loader is an abstract class. It belongs to a java.lang package. It loads classes from different resources. Java class loader is used to load the classes at run time. In other words, JVM performs the linking process at runtime. In this way, uniqueness is maintained in the runtime environment. It is essential to execute a Java program.

Java Class Loader is based on three principales :-

- Delegation principle -
- Visibility principle
- Uniqueness principle.

Types of Class Loader -

In Java, every class loader has a predefined location from where they load class files. There are following types of class loader in Java:

1) Bootstrap Class Loader:
It load standard JDK class files from rt.jar and other core classes. It is a parent of all class loaders. It doesn't have any parent.

2) Extension Class Loader:
It delegates class loading request to its parents. If the loading of a class is unsuccessful, it loads classes from jre/lib/ext directory.

3) System Class Loader:
It loads application specific classes from the classPath environment variable. It is a child of exctension class loader.

How Class Loader works in Java :-

When JVM request for a class, it invokes a loadclass() method of the java.lang.classloader class by passing the fully classified name of the class. The load class() method calls for find loaderclass() method to check that the class has been already loaded or not. It is required to avoid loading the class multiple times.

If the class is already loaded, it delegates the request to parent class loader to load the class. If the ClassLoader is not finding the class, it invokes the class findclass() method to look for classes in the file system.

Visibility principle states that child ClassLoader can see the class loaded by the parent ClassLoader, but vice versa is not true.

According to uniqueness principle, a class loaded by the parent should not be loaded by child class loader again.

In short, class loader follows the following rule:

- It checks if the class is already loaded.
- If the class is not loaded, ask parent class loader to load the class.
- If parent class loader cannot load class, attempt to load it in this class loader.

When classes are loaded -

There are only two cases:

1) When the new byte code is excuted.

2) When the byte code makes a static reference to a class.

For example -> System.out