Hi friends in this video we will see the Java memory management. How Memory is managed in java. What happens when java program is executed and where does all the objects are stored.

In this image you can see that The memory area inside JVM is divided into multiple parts to store specific parts of application data.

The different divisions are

Heap area

Method area

Stack

Native method stack

Pc register.

Now we will see what is the importance of each area.

Heap-

Java heap space is used to store the objects. So when ever we create object it is stored in Heap space.

This Heap space is instantiated when we start the JVM. The default java heap space size is 1280 MB.

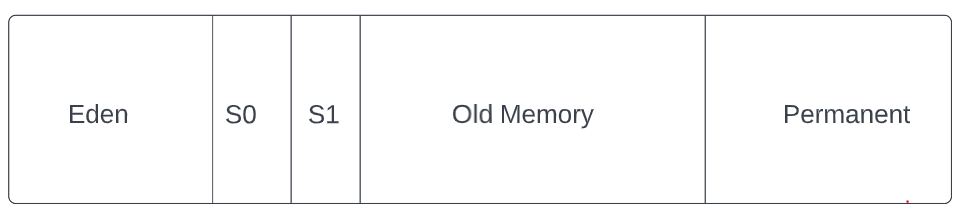
For suppose when we run the java application. It has created a lot of objects and the whole 1gb is occupied and there is no space to create a new object. Then Java will throw out of memory exception. And JVM will be terminated.

Now We may think that for some applications 1GB might not be sufficient so what we can do. For this we have a solution. we can also change this default size by setting java options.

But adding this commands in java options we can change the heap space.

-Xms<size> set initial Java heap size

-Xmx<size> set maximum Java heap size



Old Generation

Young Generation

Now what happens inside heap area. This image shows the different segments in heap area.

This heap area is divided into two parts

1. Young generation
2. Old generation

Young generation

In this image you can see the first three segments are called as young generation. SO when ever a new object is created it is first saved in this young generation. And in this young generation first it is saved in EDEN space.

When the eden space is full a process called as minor GC is triggered and all the usable objects will be moved to either S0 and eden space is freed up. And new objects will be created.

Now again when eden space is filled up, the objects from S0 will be moved to S1 and only the objects which are useful are moved and others are deleted. And objects from edenspace are moved to S0 again.

And this process continues.

Old generation

Now the objects are which are old and are staying in young generation from long time will be moved to old generation. SO from S1 the objects will be moved to old generation.

As we said minor gc will trigeered in young generation. So all the objects which are survived in GCs will be moved to old generation.

Now when this old generation is full then major GC is triggered and will try to free up some space.

We will discuss about GC in separate video.

This is how objects are stored in heap .

Stack-

Stack is the memory area in which the short lived and references to other objects in heap are stored.

Java is sequential execution program. I means it executes each line at a time. And the same program can be executed in multiple threads.

So for each thread one stack area is created. And As in java we have methods. So for each method of each thread one new block is created in stack to store local primitive values. And regerences. As soon as the method ends this the block will be useless. And used by next method.

We can provide the size of the stack, if we didn’t provide JVM will consider the default size.

This block creation we will see with below example.

Explanation of example.

Next is native method stacks-

As we discussed in previous video what are native methods . SO these are non java language which can be used BY JVM. SO this memory is same as stack but is used by native methods.