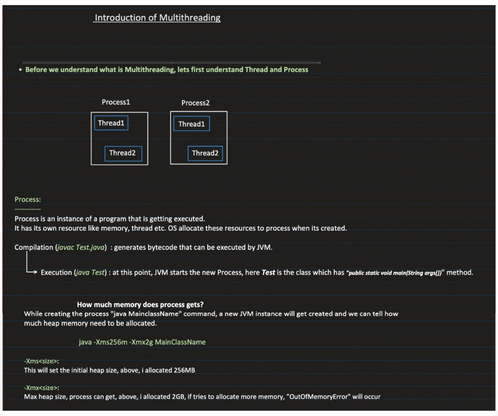
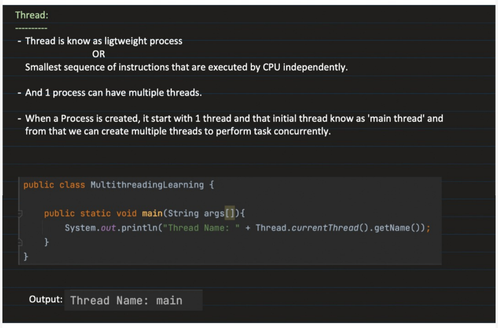
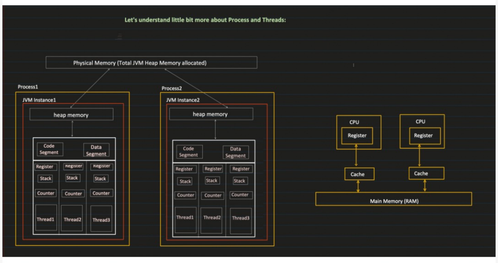
Multithreading - Part1

"Concept && Coding" YT Video Notes







Code Segment:

- Contains the compiled **BYTECODE** *(i.e machine code)* of the Java Program.
- Its read only.
- All threads within the same process, share the same code segment.

Data Segment:

- Contains the GLOBAL and STATIC variables.
- All threads within the same process, share the same data segment.
- Threads can read and modify the same data.
- Synchronization is required between multiple threads.

Heap :

- Objects created at runtime using "new" keyword are allocated in the heap.
- Heap is shared among all the threads of the same process. (but NOT WITHIN PROCESS)
  (let say in Process1, X8000 heap memory pointing to some location in physical memory, same X8000 heap memory point to differet location for Process2)
- Threads can read and modiy the heap data.
- Synchronization is required between multiple threads.

Stack:

- **Each thread has its own STACK.**
- **It manages, method calls, local variables.**

Register:

- **When JIT (Just-in time) compiles converts the Bytecode into native machine code, its uses register to optimized the generated machine code.**
- **Also helps in context switching.**
- **Each thread has its own Register.**

Counter:

- **Also know as Program Counter, it points to the instruction which is getting executed.**
- **Increments its counter after successfully exectuion of the instruction.**

All these are managed by JVM.

Definition of Multithreading:

- **Allows a program to perform multiple task at the same time.**
- **Multiple threads share the same resource such as memory space but still can perform task independently.**

Benefits and Challenges of Multithreading:

Benefits :

- **Imporved performance by task parallelism**
- **Responsiveness**
- **Resource sharing**

Challenges:

- **Concurrency issue like deadlock, data inconsistency etc.**
- **Synchronized overhead.**
- **Testing and Debugging is difficult.**

Multitasking vs Multithreading