



# Process Migration

Operating Systems Lab Project

Group No : 41  
Anshul Jethvani : 201501037  
Jay Modi : 201501059  
Meet Patel : 201501074



# Problem Definition

**Process Migration :** It refers to the mobility of executing (or suspended) processes in a distributed computing environment.

In distributed computing, processes are moved from one computing environment to another using a distributed file system and shared network, to increase effectiveness of process execution .

On multicore machines, process migration happens as a standard part of process scheduling. Its comparatively easy to migrate a process within a given machine, since most resources (memory, files, sockets) do not need to be changed, only the execution context (primarily program counter and registers) are subject to change.



# Understanding Few JAVA keywords

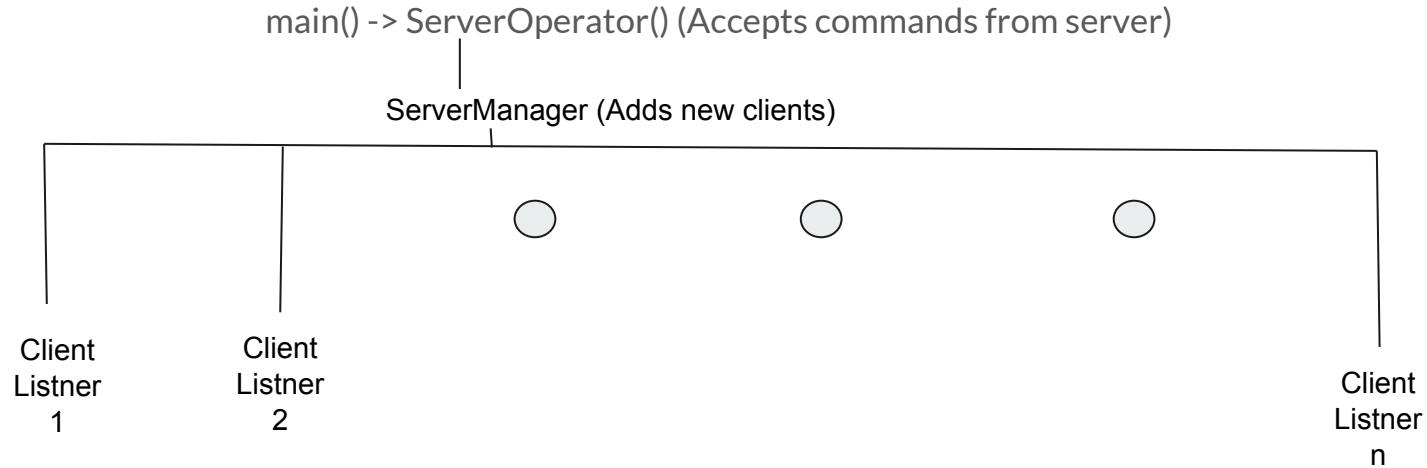
- **Synchronized** : Purpose of this keyword is to only allow one thread at a time into a particular section of code and thus allowing us to protect critical section. Synchronization happens only when we Enter/Exit the synchronized block (Critical Section).
- **Volatile** : Used to indicate that a variable's value will be modified by different threads and value of this variable will never be cached thread-locally i.e. all reads and writes will go straight to *main memory*. Synchronization happens only whenever a volatile variable is accessed.
- **Transient** : Used in serialization mainly to prevent a particular variable from being serialized. When JVM comes across transient keyword, it ignores original value of the variable and save default value of that variable data type (For Example : default 0 for int data type).
- **Atomic** : Declaring an atomic variable guarantees that operations made on the variable occur in an atomic fashion, i.e., that all of the substeps of the operation are completed within the thread they are executed and are not interrupted by other threads.



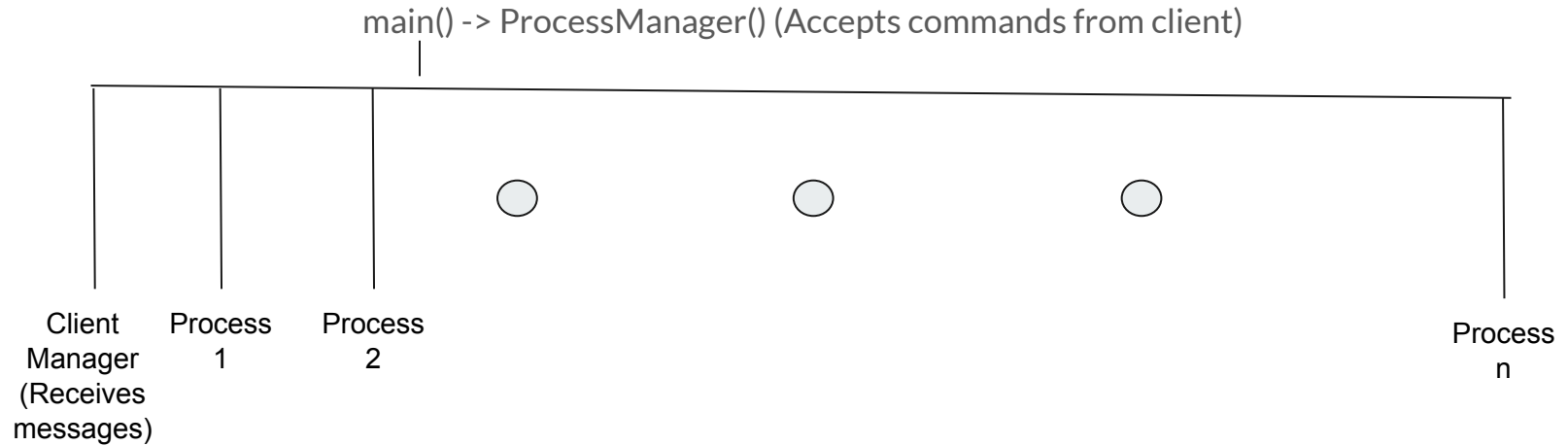
# OS Concepts used to solve the problem

- ❖ Thread Management :
  - ServerManager thread:- to accept client connections
  - ClientListener threads:- to communicate with newly joined client.
- ❖ Concurrency Mechanisms : to make the program thread safe following keywords are used :
  - Synchronized
  - Volatile
  - Transient
  - Atomic
- ❖ Exception Handling : Exceptions have been handled and appropriate error messages have been displayed wherever possible using Java's Exception Handling mechanism. This takes care of possible aborts and faults.

# Architecture: Server



# Architecture: Client





# Technical Details

- > Server can handle multiple clients
- > Each client can create any number of IOProcess as well as NonIOProcess
- > In NonIOProcess, we increase a counter variable by one at every half second.
- > In IOProcess, we read the contents of a file, sort it and write it to another file. (Paths of these files are passed as input parameters while creating a process inside a client)



Thank you...