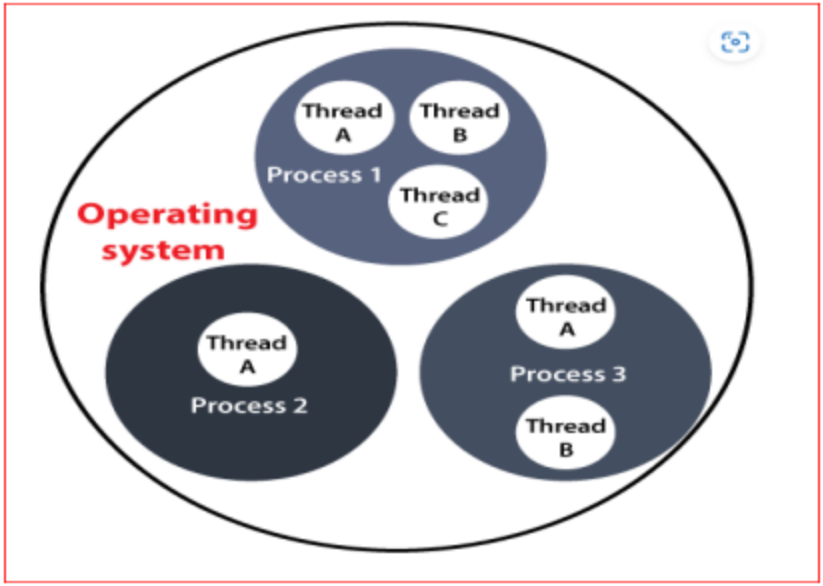
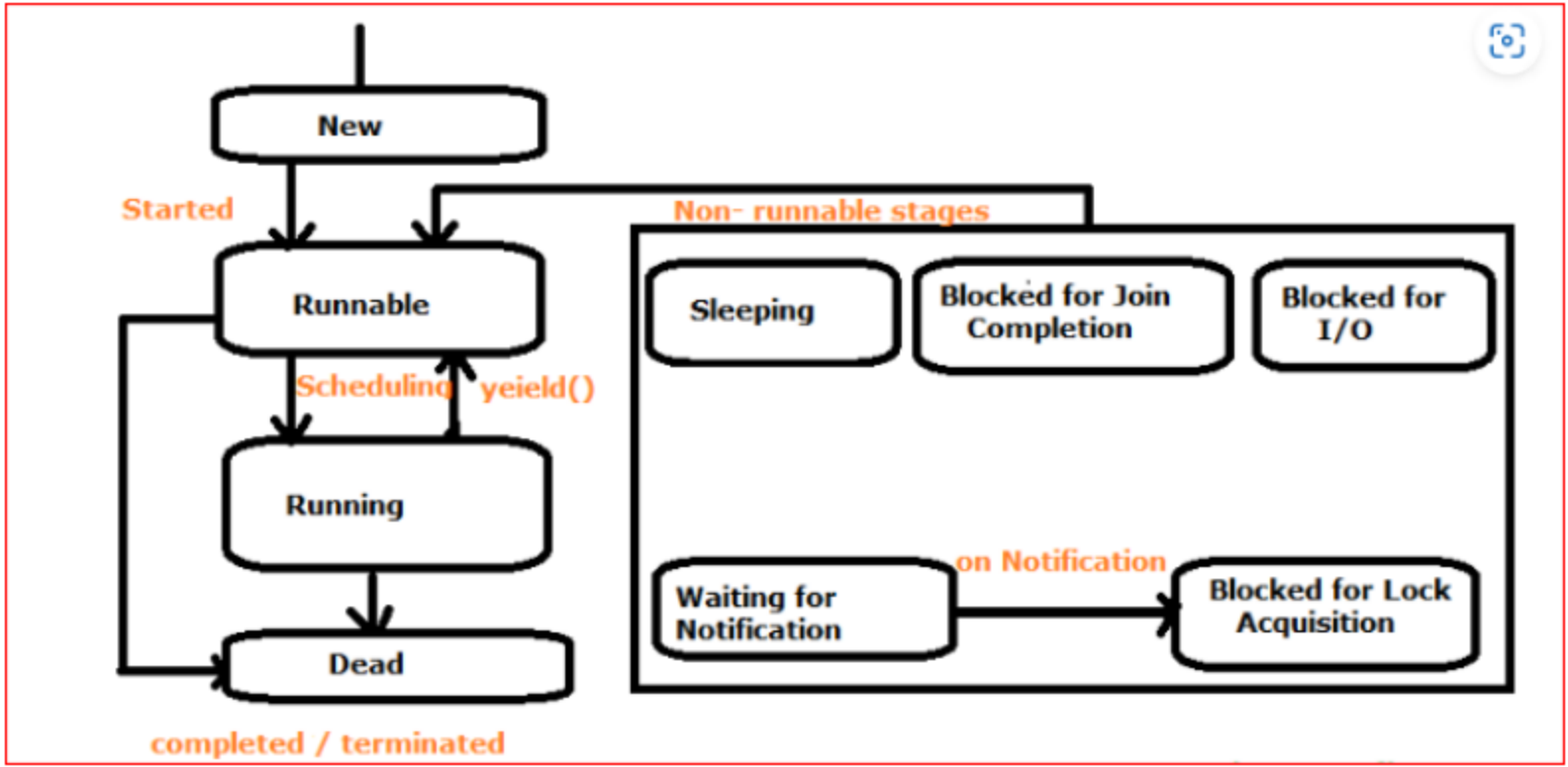
Thread

A Thread is a very light-weighted process, or we can say the smallest part of the process that allows a program to operate more efficiently by running multiple tasks simultaneously.

In order to perform complicated tasks in the background, we used the Thread concept in Java. All the tasks are executed without affecting the main program. In a program or process, all the threads have their own separate path for execution, so each thread of a process is independent.



Life Cycle


New
Started
Runnable
Scheduling
yeield()
Running
Dead
completed / terminated
Non- runnable stages
Sleeping
Blocked for Join Completion
Blocked for I/O
Waiting for Notification
on Notification
Blocked for Lock Acquisition

1.New State:

A thread has been created but not started yet. A thread will be started by calling its start() method.

2.Runnable State:

This state is also called the ready to run stage also called queue. A thread starts in a runnable state by calling start() method.
The Thread scheduler decides which thread runs and how long.

3.Running State:

If a Thread is executing that means Thread is in the Running stage.

4.Dead State:

Once a Thread reaches a dead state it can not run again.

5. Non runnable States:

A Running Thread transit to one of the non runnable states, depending upon the circumstances.
A Thread remains non runnable until a special transition occurs.
A Thread does not go directly to the running state from a non-runnable state.