

Thread life Cycle

A thread in Java at any point of time exists in any one of the following states. A thread lies only in one of the shown states at any instant:

1. New
2. Runnable
3. Blocked
4. Waiting
5. Timed Waiting
6. Terminated

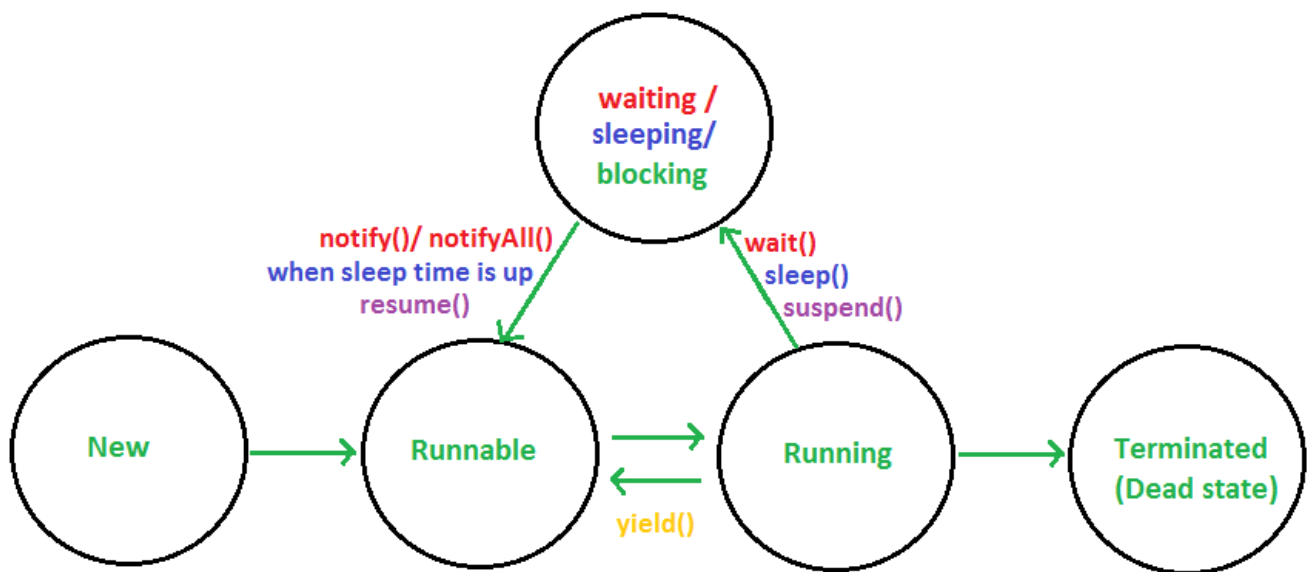


Fig. THREAD STATES

Life Cycle of a thread

1. **New Thread:** When a new thread is created, it is in the new state. The thread has not yet started to run when the thread is in this state. When a thread lies in the new state, its code is yet to be run and hasn't started to execute.
2. **Runnable State:** A thread that is ready to run is moved to a runnable state. In this state, a thread might actually be running or it might be ready to run at any instant of time. It is the responsibility of the thread scheduler to give the thread, time to run.
3. **Blocked/Waiting state:** When a thread is temporarily inactive, then it's in one of the following states:

- Blocked
 - Waiting
4. **Timed Waiting:** A thread lies in a timed waiting state when it calls a method with a time-out parameter. A thread lies in this state until the timeout is completed or until a notification is received. For example, when a thread calls sleep or a conditional wait, it is moved to a timed waiting state.
 5. **Terminated State:** A thread terminates because of either of the following reasons:
 - Because it exits normally. This happens when the code of the thread has been entirely executed by the program.
 - Because there occurred some unusual erroneous event, like segmentation fault or an unhandled exception.