Java Thread States

NEW

- `Thread t1 = new Thread();`

RUNNABLE

- `t1.start();` (NEW → RUNNABLE)

WAITING

- `Object.wait();` (RUNNABLE → WAITING)
- `notify() / notifyAll();` (WAITING → RUNNABLE)
- `Thread.join();` (RUNNABLE → WAITING)

Details on Join:

1. The thread that calls the join method goes into the WAITING state until the thread on which the `join()` was called completes execution.
2. Dies out will come the thread that calls the `join()` from WAITING → RUNNABLE state

Example:

Java
t1.join(); // The MAIN thread calls the join() method on t1 .

// Main will go into WAITING state until the t1 completes the execution.

BLOCKED

- The thread fails to acquire the lock. (RUNNABLE → BLOCKED)
- When the lock is release, the thread comes from BLOCKED → RUNNABLE

TIMED_WAITING

1. Thread.sleep(ms); // Static method (RUNNABLE → TIMED_WAITING)

1. Time expires (TIMED_WAITING → RUNNABLE)

2. Object.wait(ms); // Instance method (RUNNABLE → TIMED_WAITING)

- Time expires (TIMED_WAITING → RUNNABLE)

- `notify()`/ `notifyAll()` is called (TIMED_WAITING → RUNNABLE)
- *whatever happens first*

3. Thread.join(ms); // Instance method (RUNNABLE → TIMED_WAITING)

- Time expires (TIMED_WAITING → RUNNABLE)
- Dies out (TIMED_WAITING → RUNNABLE)
- *Whatever happens first*

TERMINATED

- When the `run()` completes the execution, the thread goes into the TERMINATED state.

Producer / Consumer (Semaphores)

Setup:

- `Full = 1` // There is 1 slot available for the producer to produce
- `Empty = 0` // There is 0 slot available for the consumer to consume.

Producer Logic:

```
Full.acquire() // Producer is going to
produce a value.
// Producer will be able to acquire a
slot if is available

Mutex.acquire(); // Producer is locking
the shared variable.
Slot = value; // Modifying the shared
variable
Mutex.release(); // Releasing the lock

Empty.release(); // Increase the
permits available for the consumer
// Now Empty = 1; // Now consumer can
consume.
```

Consumer Logic:


Empty.acquire(); // Consumer is going to consume a value
Mutex.acquire(); // acquire a lock
Int value = slot;
Slot = null;
Mutex.release(); // release the lock
Full.release(); // Increase the permits available for the producer.