MLQ - Starvation of high priority thread

1. T1 (low priority) starts, runs and acquires the lock `l`
2. T2 (medium priority) starts, preempts the CPU and runs
3. T3 (high priority) starts, preempts the CPU, runs
   but gets blocked while trying to acquire the lock `l`
4. T2 is elected to run (highest priority thread to be ready to run)

◉ **Problem** : starvation of a high priority thread

✓ **Solution** : priority donation

MLQ - Priority donation (simple example)

1. T1 (low priority) starts, runs and acquires the lock `l`
2. T2 (medium priority) starts, preempts the CPU and runs
3. T3 (high priority) starts, preempts the CPU, runs
   but gets blocked while trying to acquire the lock `l`
4. T3 gives its high priority to T1
5. T1 (now high priority) runs, releases the lock and returns to low priority immediately after
6. T3 (now unblocked) preempts the CPU and runs