Assignment 2:

2. There is a counter example, assuming there are two threads A and B wanting to get the lock, thread B comes earlier than A

thread A wants to set its label, label[a], while at the same time B is writing to label[b], so when A reads all labels, it gets label[b]=0 (an arbitrary wrong number).

When B comes to the conditional judgment when A concurrently write label[a], it gets flag[a]=true and label[a]<label[b], so B does nothing.

When A comes to condition judgment, because label[a]==label[b] (although A comes after B, but A reads label[b] wrongly) and id(a)<id(b), so A gets the lock.

In the above example, B comes first while B cannot get the lock. Instead, the late thread A gets the lock, which violates fairness.