Too Much Milk

Roommate A

3:00  Arrive home: no milk
3:05  Leave for store
3:10  Arrive at store
3:15  Leave store
3:20  Arrive home, put milk away
3:25  
3:30  

CS 111 Lecture Notes: Concurrency
## Too Much Milk

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
<th>Roommate A</th>
<th>Roommate B</th>
</tr>
</thead>
<tbody>
<tr>
<td>3:00</td>
<td>Arrive home: no milk</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3:05</td>
<td>Leave for store</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3:10</td>
<td>Arrive at store</td>
<td></td>
<td>Arrive home: no milk</td>
</tr>
<tr>
<td>3:15</td>
<td>Leave store</td>
<td></td>
<td>Leave for store</td>
</tr>
<tr>
<td>3:20</td>
<td>Arrive home, put milk away</td>
<td></td>
<td>Arrive at store</td>
</tr>
<tr>
<td>3:25</td>
<td></td>
<td></td>
<td>Leave store</td>
</tr>
<tr>
<td>3:30</td>
<td></td>
<td></td>
<td>Arrive home: too much milk!</td>
</tr>
</tbody>
</table>
if (milk == 0) {
    if (note == 0) {
        note = 1;
        buy_milk();
        note = 0;
    }
}
Still Too Much Milk

Thread A:

```java
if (milk == 0) {
    if (note == 0) {
        note = 1;
        buy_milk();
        note = 0;
    }
}
```

Thread B:

```java
if (milk == 0) {
    if (note == 0) {
        note = 1;
        buy_milk();
        note = 0;
    }
}
```
Second Attempt

Thread A:
1 if (note == 0) {
2     if (milk == 0) {
3         buy_milk();
4     }
5     note = 1;
6 }

Thread B:
1 if (note == 1) {
2     if (milk == 0) {
3         buy_milk();
4     }
5     note = 0;
6 }
Thread A:

1 noteA = 1;
2 if (noteB == 0) {
3     if (milk == 0) {
4         buy_milk();
5     }
6 }
7 noteA = 0;

Thread B:

1 noteB = 1;
2 if (noteA == 0) {
3     if (milk == 0) {
4         buy_milk();
5     }
6 }
7 noteB = 0;
Fourth Attempt

Thread A:

1  noteA = 1;
2  if (noteB == 0) {
3      if (milk == 0) {
4          buy_milk();
5      }
6  }
7  noteA = 0;

Thread B:

1  noteB = 1;
2  while (noteA == 1) {
3      // do nothing
4  }
5  if (milk == 0) {
6      buy_milk();
7  }
8  noteB = 0;