

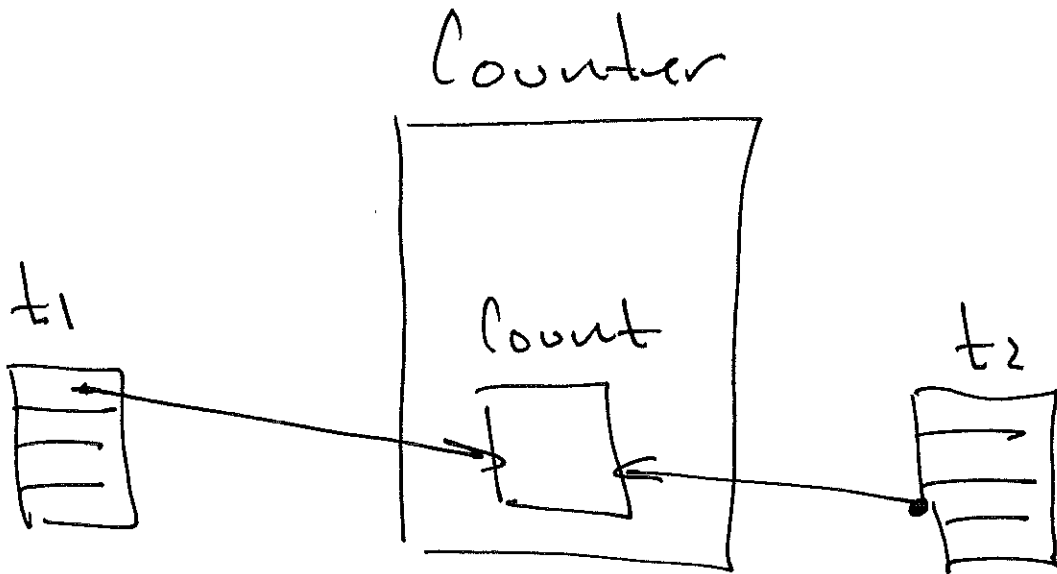
CNP > 109 3-7-11

consider check():

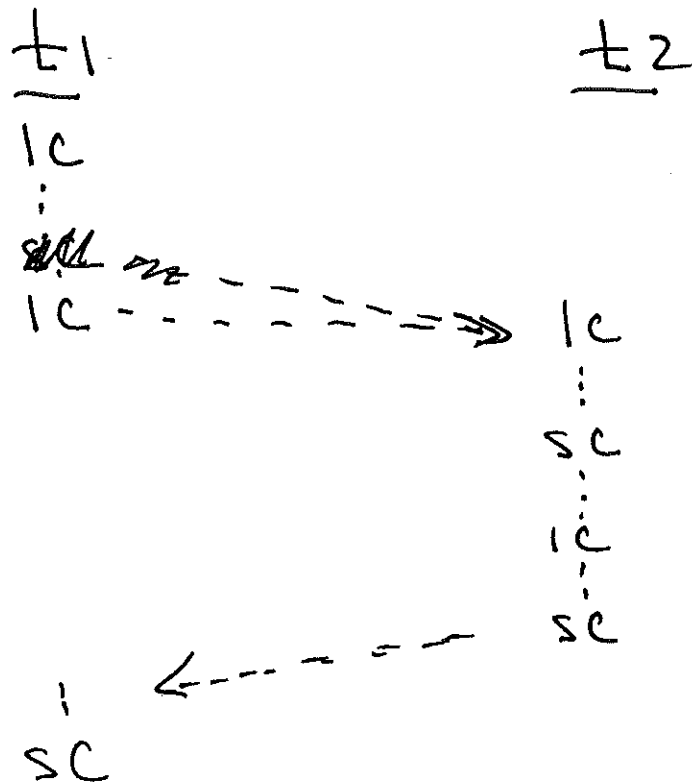
$$\text{Count} = (++\text{count}) \% \text{limit} ;$$

"assembly" code:

load count	lc
load 1	.
add	⋮
load limit	⋮
remainder	.
store count	sc



Context Switch: execution transfer
from t_1 to t_2 or t_2 to t_1



This is called race condition.

Java solves this problem through mutual exclusion, the policy of not allowing multiple threads into critical operations simultaneously.

The keyword synchronized accomplishes this:

```
synchronized void click() {
:
}
```