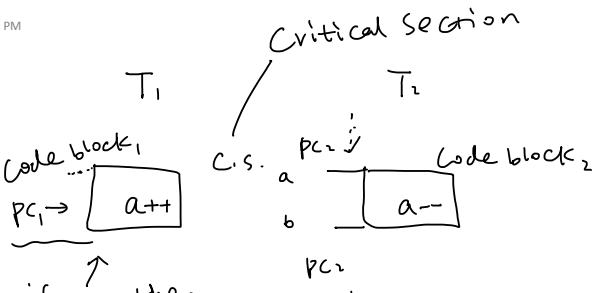
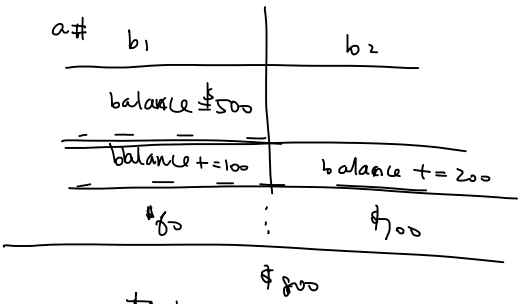
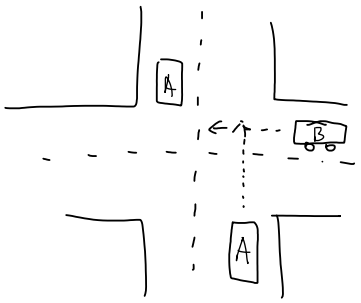




mutual exclusion ≡ only allow one thread to be in the Critical section.

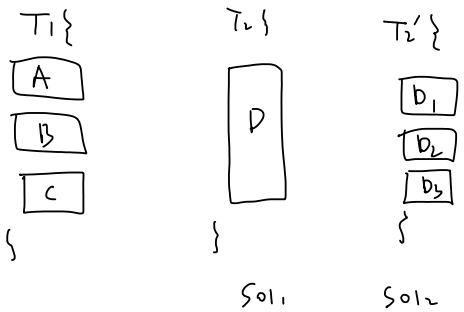
≡ instructions in the critical section are being executed on Cpu.

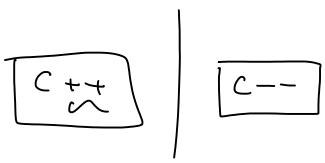


Waiting is involved.

but not for ever

⇔ progress needs.

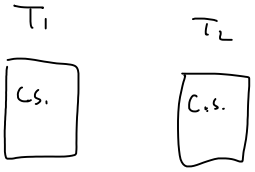




C = 5

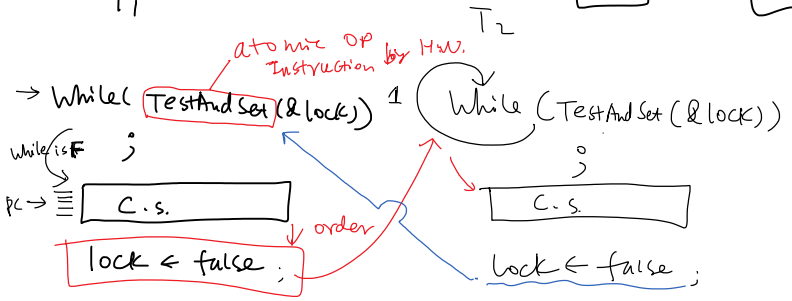
C = 4 or C = 6

ADD → ADDB



1. T1/T2 Yield() X
2. Interrupt() ←
3. Sys Call. X

lock = false



1° mutual exclusion ✓

2° progress

3° bounded waiting [duration of c.s. execution]

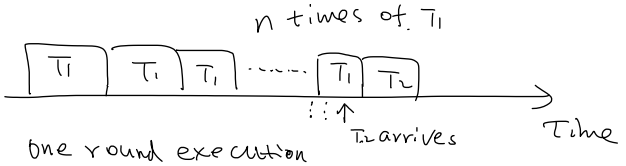
```
while (true) {
        key = TRUE;
        while ( key == TRUE)
                Swap (&lock, &key );
        // critical section
        lock = FALSE;
        // remainder section
}
```

T_1

```
while (true) {
        key = TRUE;
        while ( key == TRUE)
                Swap (&lock, &key );
        // critical section
        lock = FALSE;
        // remainder section
}
```

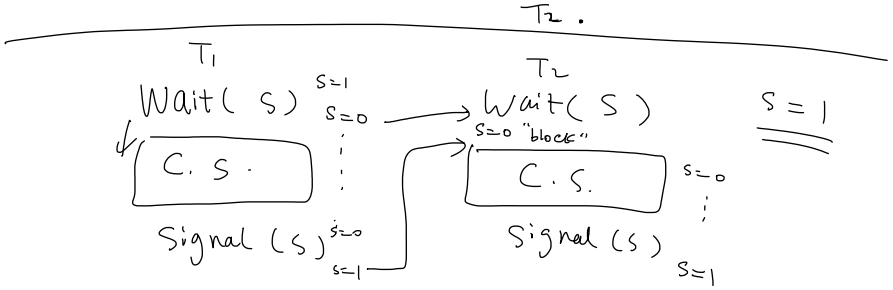
T_2

lock = False



one round execution time is C.

n.c which is not the waiting time for T_2.



Wait(S){

if S<=0

wait;

else S--;

}

Signal(S){

S++

}