

---

---

---

---

---

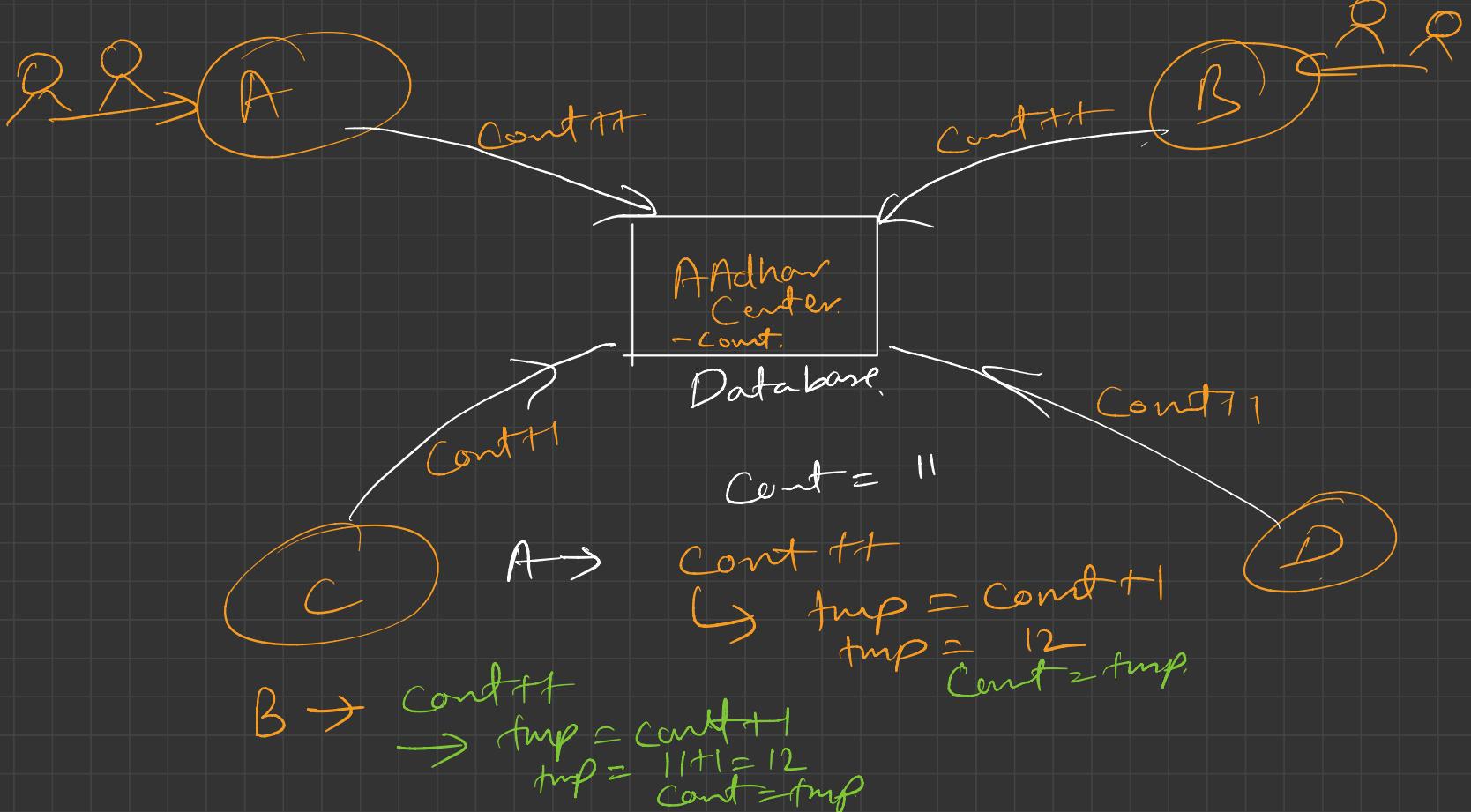


# Lec-16

Count ++ → Count = Count + 1

tmp = Count + 1

Count = tmp.





# Critical section problems

→ Solution of Race cond.

①

(Cont'd)  $\text{tmp} \leftarrow \text{cont} + 1$  ✓

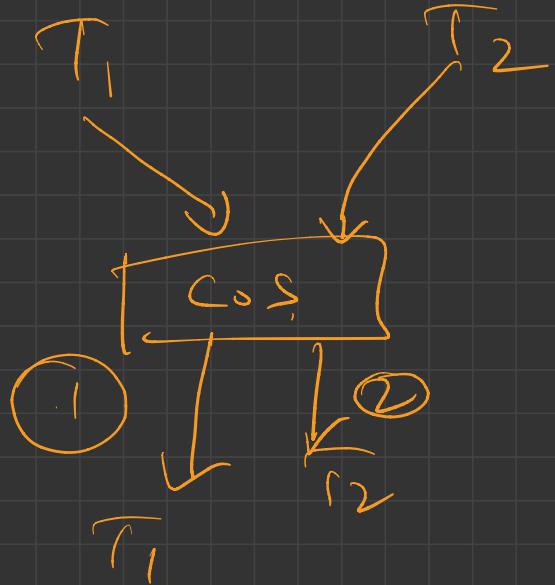
$\text{cont} = \text{tmp}$  ✓

Atomic → ✓

C++ → atomic < int > →

②

## Mutual Exclusion.



$$T_1 \rightarrow T_2$$

OR

$$\underline{T_2} \rightarrow \underline{T_1}$$

(Locks)  $\rightarrow$  Mutual Exclusion



Q → Can we use single flag ??

Q → Sol<sup>n</sup> of C.S should have 3 conditions?

① Mutual exclusion.

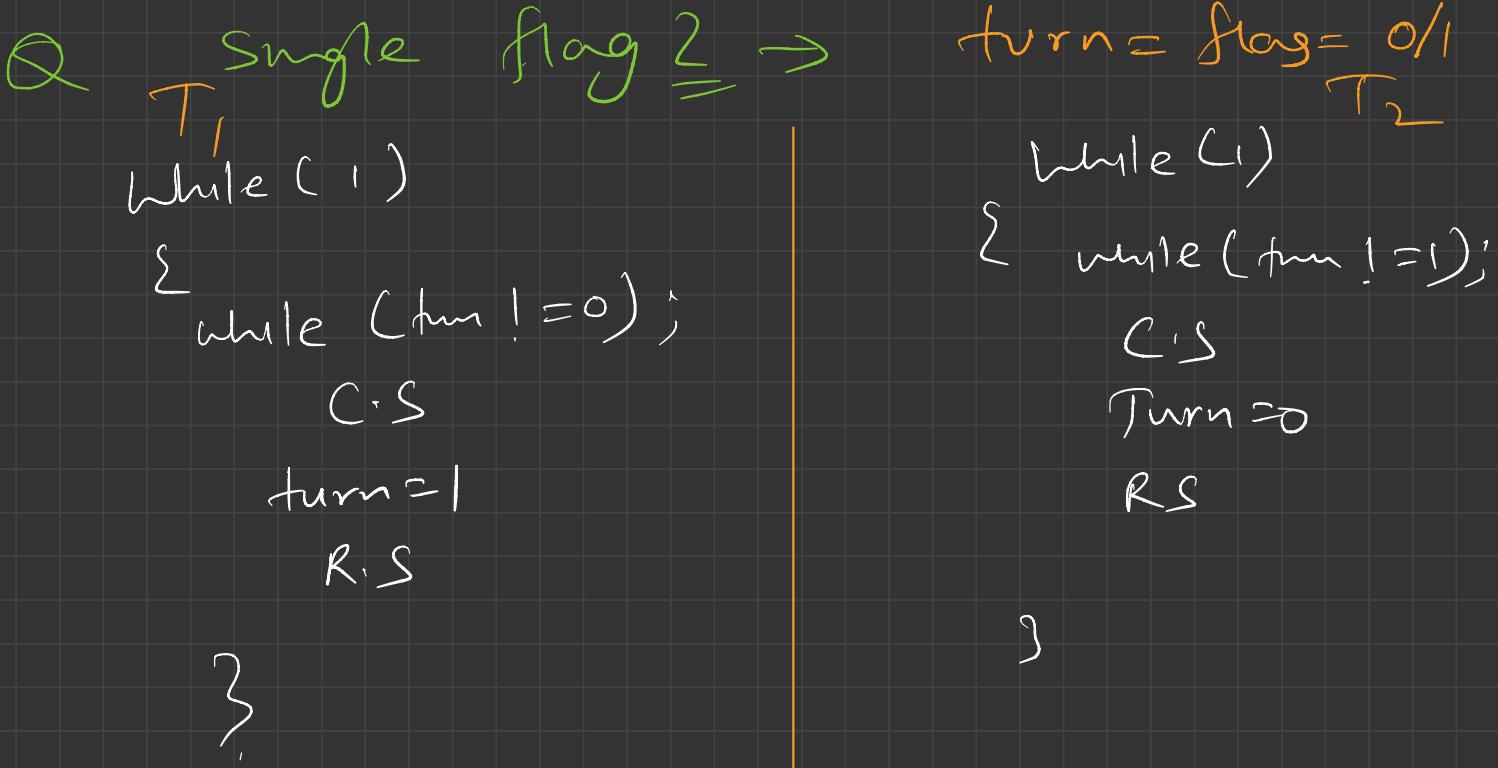
② Progress,  $T_1$        $T_2$

CS

$T_1 \rightarrow T_2 X$

$T_2 \rightarrow T_1 X$

③ Bonded waiting  
→ indefinite waiting X.  
→ Limited waiting time.

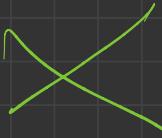


①  $\rightarrow$  turn = 0  $\rightarrow$   $T_1 \rightarrow T_2$ ,  
 $turn = 1 \Rightarrow T_2 \rightarrow T_1$

②

Progress — fixed order.

=



→ Single flag X



Implementation

→ Peterson's soln.

\* Peterson's Solution —  $\underline{\text{flag[2]}}$  →  
— turn

$\text{flag[2]}$  → indicate if a thread is ready to  
enter the CS,  $\text{flag[i]} = \text{true}$  implies that  
 $p_i$  is ready

turn → indicates whose turn is to enter the  
CS.

↳ Q1

$T_1$

while(1)

{  
  flag[0] = T

  turn = 1

  while (turn == 1 && flag[1] == F),

C.S

  flag[0] = F

}

Flag [ ]

turn :

$T_2$

while(1)

{  
  flag[1] = T

  turn = 0

  while (turn == 0 && flag[0] == T),

C.S

  flag[1] = F

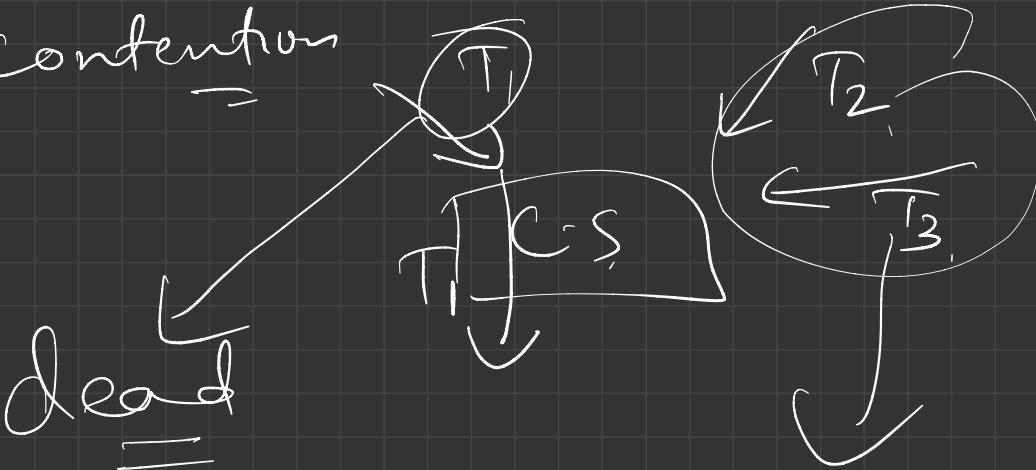
}

Doubt gen ✓.



# Locks disadvantages

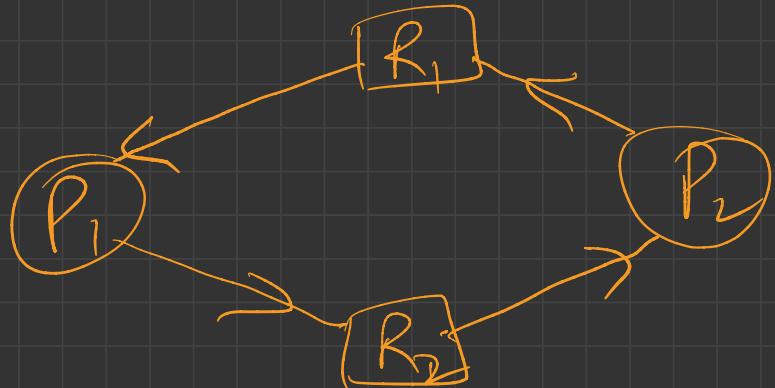
## ① Contention



dead  
=

Infinite wait.

② Deadlock.



③ Debugging issue

④ Starvation ↴

