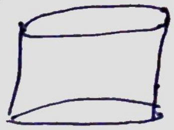
LEC - 19 | Reader-writer problem

- Synchronization karni h ki ek baar mae ek hi write write kr ske but multiple reader read kr ske.



DB
shared resource

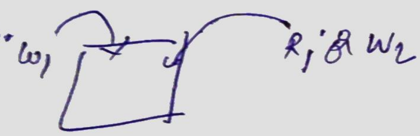
① Reader thread → Read

② writer thread → write/update

→ If >1 readers are reading
⇒ No issue.

→ If >1 writers OR 1 writer & some other thread (R/W) Parallel;
then Race cond" & data inconsistent.

Prob.



Ex - writer W1 —write→ ABCD
R1 —reads→ ABCD
} happening at same time

but writer ne thoder mae
ABCD ⟶ ABZJ kar diya th
reader ne inconsistent data read ki;

Sol^{n} — Using Semaphore

① mutex – binary semaphore.
- to ensure mutual exclusion, when read count (RC) is updated.

⇒ No two threads modify RC at same time.

② wrt → binary sema.
- common for both readers & writers.

③ read count (RC) → integer {0} → initialize
→ (not sema)
- tracks how many readers are reading in CS.

* writer Sol^{n}

```
do {
    wait (wrt)
    // do write operation
   signal (wrt);
} while (true);
```

* Reader Sol^{n}

```
do{
wait (mutex); // to mutex read Count variable
RC ++;
if (RC ==1)
wait (wrt); // ensures no writer can
            // enter if there is even
            // one reader
signal (mutex);
// C.S : Reader is reading
wait (mutex)
 RC --; // a reader leaves
if (RC==0) // no reader is left in CS.
 signal (wrt); // writer can enter
 signal (mutex); // reader leaves
} while (1).
```