Reader - Writer Problem

- There is a database to be shared among multiple Concurrent processes.
- Reader: Some of the processes may only want to read the database.
- Writer: Some of the processes may want to update the database (read + write)
- If two readers access the database concurrently, no issues arise.
- If a writer and some other processes (reader or writer) access the database concurrently, there can be adverse effects.

→ 1 1
→ 2 2 ←
→ 3 3

1000 1,
1000
1000
3000

× write + reader

Example :-

Two people having joint bank account depositing money at the same time.

```
→ bal = get_balance() ← database
→ bal = bal + deposited
→ update_balance(bal) → db
```

writer

①	②
1000	1000
2000	2000
2000	2000

Solution using mutex and Semaphore

- writers should have exclusive access to the critical section code while updating.
- First readers-writers problem ← 1 Reader

  writers only get access to shared database when there are no readers waiting. writers
- Second readers-writers problem

  Readers only get access to shared database when there are no writers waiting. Readers
- A solution to either problem may result in starvation. How?

Solution to first readers-writers problem

```
Mutex mutex; // mutual exclusion for variable readcount
Semaphore wrt = 1; // mutual exclusion + signalling}
int readcount = 0; // no of processes currently reading
```

Reader, writer

Writer

```
do {
    wait(wrt);    wrt < 0
    // update
    signal(wrt);
} while (True);
```

1 , 0

N ,

Reade , wrt

1 , wrt

Reader

```
do {
    wait(mutex);
    readcount++;
    if (readcount == 1) {        1 reader
        wait(wrt);               if (wrt < 0)
    }
    signal(mutex);
    // reading is performed
    wait(mutex);
    readcount--;
    if (readcount == 0) {
        signal(wrt);
    }
    signal(mutex);
} while (True);
```