# CMP 3001

Reader-writer problem is an example of the synchronization.

-Multiple reader allowed to read at the same time.

-Only one writer can write at the same time.

-İf it is in the process of reading,it is forbidden to write.

- If a writer has begun writing process, then No reader is allowed to read.

Semaphore forWriteS= This Semaphore for writing

Semaphore forReadS= This Semaphore for reading

```
public void readLock()
```

- If 1 or more readers are reading, then Other readers may read as well and No writer may perform write function until all readers have finished reading
- I used “forWriteS.acquire()” for no writer may perform write function until all readers have finished reading.
- I used “forReadS.release()” at end.

```
public void readUnLock()
```

1-forReadS.acquire();

2- to reduce count of readers

3-if (countOfReader == 0) forWriteS.release() because writer can write if any readers aren’t reading.

4- forReadS.release();

```
public void writeLock()
```

Only one writer can write at the same time therefore I used acquire for Semaphore for writing.

`public void writeUnLock()`

I Released write lock with “forWriteS.release()”.

Console output example:

4 writing

4 is done writing.

4 reading, count of reader = 1

4 is finish reading, count of reader = 0

3 writing

3 is done writing.