SSE2033

SUNGKYUNKWAN UNIVERSITY

Fall 2019

System Software Experiment 2
Thread Synchronization

Prof. Jinkyu Jeong (*jinkyu@skku.edu*)
TA – Gyusun Lee (*gyusun.lee@csi.skku.edu*)
TA – Jiwon Woo (*jiwon.woo@csi.skku.edu*)
Computer Systems and Intelligence Laboratory (*http://csi.skku.edu*)
Sungkyunkwan University

Example 1

- Make program
  - 5 Reader + 1 Writer
  - Writer updates value 1,000,000 times
  - Each reader reads value 10,000,000 times

- Use pthread_mutex

Example 2

- Implement same thing using pthread_spinlock
  - pthread_spinlock_t s
  - pthread_spin_init(&s, PTHREAD_PROCESS_PRIVATE)
  - pthread_spin_[un]lock(&s)
  - pthread_spin_destroy(&s)

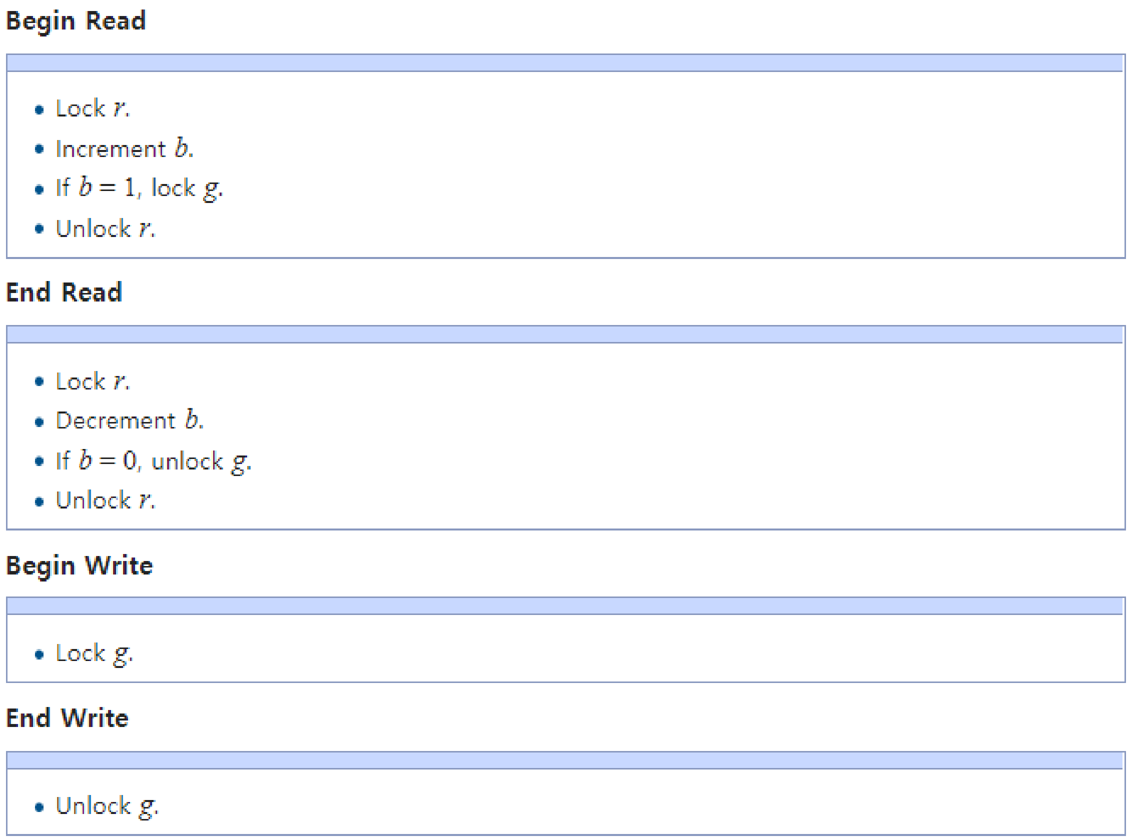
- What is the difference?

Readers-Writer Lock

- Reader blocks other reader
  - Do we need this?
- Implementation
  - Using two mutexes
  - Using a condition variable and a mutex

Readers-Writer Lock(1)

- Two mutex



Begin Read

- Lock r.
- Increment b.
- If $b = 1$, lock g.
- Unlock r.

End Read

- Lock r.
- Decrement b.
- If $b = 0$, unlock g.
- Unlock r.

Begin Write

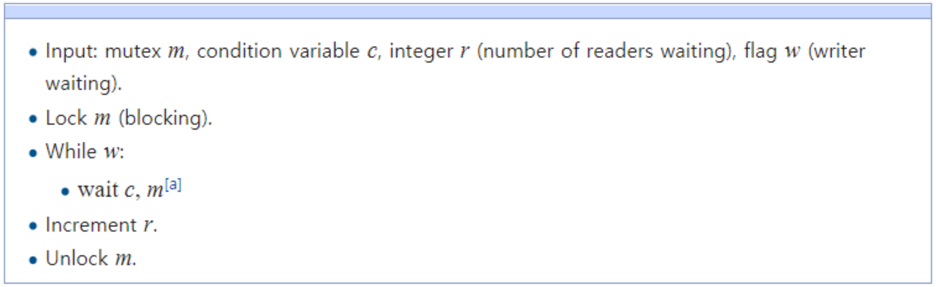
- Lock g.

End Write

- Unlock g.

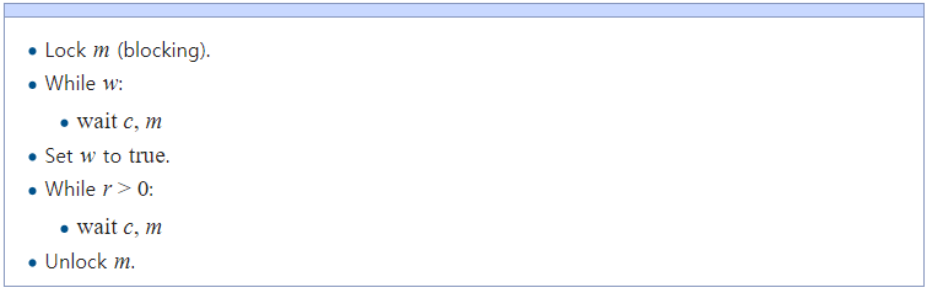
Readers-Writer Lock(2)

- A condition variable and a mutex
  - Lock for read



```
• Input: mutex $m$, condition variable $c$, integer $r$ (number of readers waiting), flag $w$ (writer waiting).
• Lock $m$ (blocking).
• While $w$:
    • wait $c$, $m$[a]
• Increment $r$.
• Unlock $m$.
```

  - Lock for write



```
• Lock $m$ (blocking).
• While $w$:
    • wait $c$, $m$
• Set $w$ to true.
• While $r > 0$:
    • wait $c$, $m$
• Unlock $m$.
```

Example 3-4

- Example 3
  - Implement program with readers-writer lock(Using mutex)
- Example 4
  - Implement program with readers-writer lock(Using condition variables)
- What is the difference?