

Modelos para Computação Concorrente ou Sistemas Operacionais

Memória Compartilhada –
Semáforos – Readers and Writers

(com slides de Ben-Ari)

Fernando Luís Dotti

Bibliografia Base

[disponível na biblioteca]

M. Ben-Ari

Principles of Concurrent and Distributed Programming

Second Edition

Addison-Wesley, 2006

© Mordechai Ben-Ari 2006

Readers and Writers

- Acesso concorrente de leitura e escrita a um recurso
- Readers and writers
 - Enquanto processos readers estiverem lendo, outros readers podem ler
 - Writers tem que escrever quando nao houverem processos lendo ou escrevendo

Leitores e escritores

- vários leitores (threads) : NL
 - cada leitor fica em loop lendo
 - acessam concorrentemente (entre leitores) a memória compartilhada
- vários escritores (threads) : NE
 - cada escritor fica em loop escrevendo
 - tem que ter acesso exclusivo (sozinho) acessando a area compartilhada

Leitores e escritores

- M: memória compartilhada
- NL leitores
 - loop
 - controle
 - le(M)
 - controle
- NE escritores
 - loop
 - controle
 - escreve(M)
 - controle

Leitores e escritores

- como permitir leitores concorrentes na leitura, mas, quando escritor precisa acesso, tem que ser exclusivo ?

Leitores e escritores

little book of semaphores

Listing 4.13: Readers-writers initialization

```
1 int readers = 0
2 mutex = Semaphore(1)
3 roomEmpty = Semaphore(1)
```

Listing 4.14: Writers solution

```
1 roomEmpty.wait()
2     critical section for writers
3 roomEmpty.signal()
```

Listing 4.15: Readers solution

```
1 mutex.wait()
2     readers += 1
3     if readers == 1:
4         roomEmpty.wait()    # first in locks
5 mutex.signal()
6
7 # critical section for readers
8
9 mutex.wait()
10    readers -= 1
11    if readers == 0:
12        roomEmpty.signal() # last out unlocks
13 mutex.signal()
```