

## UFCG UASC/CEEI

### Disciplina: Programação Concorrente Professor: Thiago Emmanuel Pereira Prova 2

Q1 - Um estrutura de dados, usada por múltiplas threads, pode usar uma trava de leitura-escrita para implementar a semântica de, ao ser modificada por uma thread, evitar que outras thread usem a estrutura concorrentemente (seja para leitura ou escrita). Isso implica nas seguintes restrições:

- a) Qualquer número de threads de leitura pode estar na região crítica concorrentemente;
- b) Threads de escrita precisam ter acesso exclusivo à região crítica.

Uma solução para esse problema usando semáforos é dado no livro LBoS

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

//controle para threads de escrita
roomEmpty.wait()
    //região crítica de escrita
roomEmpty.signal()

//controle para threads de leitura
mutex.wait()
    readers += 1
    if readers == 1:
        roomEmpty.wait()# first in locks
mutex.signal()
    #região crítica de leitura
mutex.wait()
    readers -= 1
    if readers == 0:
        roomEmpty.signal() #last out unlocks
mutex.signal()
```

Re-implemente essa solução considerando monitores em java (synchronized, wait, notify, notifyAll são importantes) ou variáveis condicionais. Considere que a estrutura usada por várias threads é um objeto com a assinatura abaixo. **contains** será usado por threads de leitura enquanto **add** por threads de escrita. Você não precisa implementar a lógica de add e contains, somente o controle de concorrência. Ainda, adicione synchronized na assinatura dos métodos conforme for necessário.

```
public interface Data {
    public boolean contains(String value);
    public void add(String value);
}
```

Q2 - Explique como travas TTAS podem ter desempenho melhor que travas TAS. Sua explicação deve considerar aspectos de arquitetura de computadores.

Q3 - Discuta os trade-offs envolvidos no uso de travas que bloqueiam os processos/threads e travas que fazem busy-wait. Dê exemplos realistas de sistemas em que o uso de uma ou de outra possa fazer sentido.