

UNIVERSIDADE FEDERAL DE VIÇOSA
INF310 – PROGRAMAÇÃO CONCORRENTE E DISTRIBUÍDA
Lista de Exercícios 2

1. As operações Down (ou wait) e Up (ou post) do semáforo devem ser atômicas? Justifique.
2. Por que o mecanismo de monitor é considerado de nível mais alto que os semáforos?
3. Considerando que as threads A e B são concorrentes, atualize o código abaixo utilizando semáforos para garantir que, a qualquer momento, $0 \leq x \leq 10$.
Obs: não é permitido inserir ou alterar as instruções que alterem o valor de x.

```
int x = 0; //variável global
```

```
void thread_A() {  
    while(true) {  
        x = x+1;  
    }  
}
```

```
void thread_B() {  
    while(true) {  
        x = x-1;  
    }  
}
```

4. Considerando o problema do barbeiro dorminhoco (implementação vista em aula utilizando semáforos), escreva uma solução utilizando monitor. Você deve apresentar uma possível implementação para as funções **cliente_chega** e **barbeiro_faz_corte** do monitor **barbearia**, de acordo com o esboço das threads mostradas abaixo. Perceba que a condição de parada do barbeiro é que todos os clientes tenham chegado e que tenham ou sido atendidos ou desistido de esperar. Uma possibilidade é que a função **barbeiro_faz_corte** retorne **false** quando as threads clientes terminarem.

```
void barbeiro() {  
    while (<condição_de_parada>)  
        barbearia.barbeiro_faz_corte();  
}
```

```
void cliente(int id) {  
    barbearia.cliente_chega(id);  
}
```

5. Considere o problema do jantar dos canibais descrito a seguir: Em uma tribo de canibais, os canibais passam o tempo comendo pedaços de missionários que são servidos pelo cozinheiro da tribo. Ao tentar comer, se não houverem mais pedaços, um canibal acorda o cozinheiro. O cozinheiro cozinha mais alguns pedaços e volta a dormir.
Implemente esse problema considerando a sincronização entre uma thread cozinheiro e várias threads canibais utilizando semáforos. Atenção para os seguintes requisitos obrigatórios:
 - a) O cozinheiro não deve ser acordado se ainda houverem pedaços disponíveis.
 - b) Os canibais só voltam a comer depois que todos os pedaços foram servidos.
 - c) Cada canibal come um pedaço de cada vez mas, se for rápido, pode comer vários pedaços em seguida antes que outros comam.