

Avaliação 5 – Coordenação de Processos – região crítica; condições de disputa e espera ocupada

Fonte: Prof. Msc. Jacson Rodrigues; Maziero, 2019.

Responda as questões **justificando** sua solução.

1. Explique o que são condições de disputa, mostrando um exemplo real.
2. Sobre as afirmações a seguir, relativas aos mecanismos de coordenação, indique quais são incorretas, justificando sua resposta:
 - a) A estratégia de inibir interrupções para evitar condições de disputa funciona em sistemas multi-processados.
 - b) Os mecanismos de controle de entrada nas regiões críticas proveem exclusão mútua no acesso às mesmas.
 - c) Os algoritmos de *busy-wait* se baseiam no teste contínuo de uma condição.
 - d) Condições de disputa ocorrem devido às diferenças de velocidade na execução dos processos.
 - e) Condições de disputa ocorrem quando dois processos tentam executar o mesmo código ao mesmo tempo.
 - f) Instruções do tipo *Test&Set Lock* devem ser implementadas pelo núcleo do SO.
 - g) O algoritmo de Peterson garante justiça no acesso à região crítica.
 - h) Os algoritmos com estratégia *busy-wait* otimizam o uso da CPU do sistema.
 - i) Uma forma eficiente de resolver os problemas de condição de disputa é introduzir pequenos atrasos nos processos envolvidos.
3. Explique o que é espera ocupada e por que os mecanismos que empregam essa técnica são considerados ineficientes.
4. Em que circunstâncias o uso de espera ocupada é inevitável?
5. Considere **ocupado** uma variável inteira compartilhada entre dois processos A e B (inicialmente, **ocupado** = 0). Sendo que ambos os processos executam o trecho de programa abaixo, explique em que situação A e B poderiam entrar simultaneamente nas suas respectivas regiões críticas.

```
1 while (true) {  
2     regioao_ao_critica();  
3     while (ocupado) {};  
4     ocupado = 1;  
5     regioao_critica();  
6     ocupado = 0;  
7 }
```