

BCC34G – Sistemas Operacionais

Lista de Exercícios #05 – Sincronização e Concorrência

1. Explique o que é uma seção crítica.
2. Explique e exemplifique uma condição de disputa (*race condition*).
3. Explique os três requisitos para uma solução do problema de seção crítica: exclusão mútua, progresso, espera limitada. ^[1]
4. Explique o conceito de espera ocupada (*busy waiting*)? ^[1]
5. Explique por que spinlocks não são apropriados para sistemas com um único processador, mas são usados com frequência em sistemas com múltiplos processadores. ^[1]
6. Explique por que as interrupções não são apropriadas para a implementação de operações de sincronização em sistemas multiprocessadores. ^[1]
7. Mostre que, se as operações de semáforo *wait()* e *signal()* não forem executadas atomicamente, a exclusão mútua pode ser violada. ^[1]
8. Mostre como um semáforo binário pode ser usado para implementar a exclusão mútua entre n processos. ^[1]
9. O algoritmo de Dekker foi a primeira solução correta para o problema da seção crítica envolvendo dois processos. Explique o pseudo código e responda: O algoritmo garante a exclusão mútua? O algoritmo é livre de deadlock? O algoritmo é livre de starvation?

```

variables
  wants_to_enter : array of 2 booleans
  turn : integer

wants_to_enter[0] ← false
wants_to_enter[1] ← false
turn ← 0 // or 1

p0:
  wants_to_enter[0] ← true
  while wants_to_enter[1] {
    if turn ≠ 0 {
      wants_to_enter[0] ← false
      while turn ≠ 0 {
        // busy wait
      }
      wants_to_enter[0] ← true
    }
  }
  // critical section
  turn ← 1
  wants_to_enter[0] ← false
  // remainder section

p1:
  wants_to_enter[1] ← true
  while wants_to_enter[0] {
    if turn ≠ 1 {
      wants_to_enter[1] ← false
      while turn ≠ 1 {
        // busy wait
      }
      wants_to_enter[1] ← true
    }
  }
  // critical section
  turn ← 0
  wants_to_enter[1] ← false
  // remainder section

```

Figura 1: Pseudocódigo do algoritmo de Dekker (Fonte: Wikipedia)

10. O kernel do Linux tem uma política de que um processo não pode manter um spinlock enquanto tenta adquirir um semáforo. Explique por que essa política foi definida. ^[1]
11. Como a instrução TSL (*Test and Set Lock*) pode ser usada para resolver o problema da seção crítica? ^[3]

TSL RX,LOCK: lê o conteúdo da variável LOCK para o registrador RX e armazena em LOCK um valor diferente de 0.
12. Em relação aos mecanismos de coordenação: ^[4]
 - I. Instruções do tipo *Test and Set Lock* devem ser implementadas pelo núcleo do SO.
 - II. O algoritmo de Peterson garante justiça no acesso à região crítica.
 - III. Os algoritmos com estratégia *busy-wait* otimizam o uso da CPU do sistema.

BCC34G – Sistemas Operacionais

Lista de Exercícios #05 – Sincronização e Concorrência

IV. Uma forma eficiente de resolver os problemas de condição de disputa é introduzir pequenos atrasos nos processos envolvidos.

V. Um semáforo é composto por um contador inteiro e uma fila de processos suspensos.

As asserções corretas são:

- (a) I, III (b) I, V (c) II, V (d) I, IV (e) III, IV

Justifique as afirmações julgadas erradas (Assim: VII está errada porque ...).

13. Em que situações um semáforo deve ser inicializado em 0, 1 ou $n > 1$? ^[4]

14. Explique cada uma das quatro condições necessárias para a ocorrência de impasses. ^[4]

15. Na prevenção de impasses: ^[4]

- (a) Como pode ser feita a quebra da condição de posse e espera?
 (b) Como pode ser feita a quebra da condição de exclusão mútua?
 (c) Como pode ser feita a quebra da condição de espera circular?
 (d) Como pode ser feita a quebra da condição de não-preempção?

16. Como pode ser detectada a ocorrência de impasses, considerando disponível apenas um recurso de cada tipo? ^[4]

17. Uma vez detectado um impasse, quais as abordagens possíveis para resolvê-lo? Explique-as e comente sua viabilidade. ^[4]

18. Em relação aos impasses: ^[4]

I. As condições necessárias para a ocorrência de impasses são: exclusão mútua, posse e espera, não-preempção e espera circular.

II. A condição de não-preempção indica que os processos envolvidos no impasse devem ser escalonados de forma não-preemptiva.

III. A condição de não-preempção pode ser detectada graficamente, no grafo de alocação de recursos.

IV. A detecção e recuperação de impasses é bastante usada, pois as técnicas de recuperação são facilmente aplicáveis.

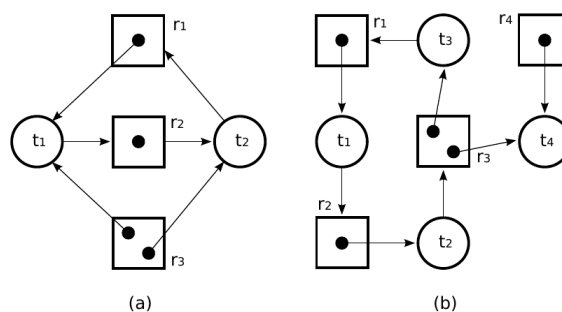
V. A condição de exclusão mútua pode ser quebrada através do uso de processos gerenciadores de recursos ou de áreas de spool.

As asserções corretas são:

- (a) II (b) I, V (c) I, III (d) III, IV (e) II, V

Justifique as afirmações julgadas erradas (Assim: VII está errada porque ...):

19. Nos grafos de alocação de recursos da figura a seguir, indique o(s) ciclo(s) onde existe um impasse: [4]



20. Um sistema tem dois processos e três recursos idênticos. Cada processo precisa de um máximo de dois recursos. Um impasse é possível? Explique a sua resposta.

BCC34G – Sistemas Operacionais

Lista de Exercícios #05 – Sincronização e Concorrência

21. Um sistema tem quatro processos e cinco recursos alocáveis. A alocação atual e as necessidades máximas são as seguintes: ^[3]

	<i>Alocado</i>	<i>Máximo</i>	<i>Disponível</i>
Processo A	10211	11213	00x11
Processo B	20110	22210	
Processo C	11010	21310	
Processo D	11110	11221	

Qual é o menor valor de x para o qual esse é um estado seguro?

22. Um estudante de Ciência da Computação designado para trabalhar com impasses pensa na seguinte maneira brilhante de eliminar os impasses. Quando um processo solicita um recurso, ele especifica um limite de tempo. Se o processo bloqueia porque o recurso não está disponível, um temporizador é inicializado. Se o limite de tempo for excedido, o processo é liberado e pode executar novamente. Se você fosse o professor, qual nota daria a essa proposta e por quê? ^[3]
23. Explique a diferença entre impasse, livelock e inanição. ^[3]

Referências:

- [1] SILBERSCHATZ, A.; GALVIN, P. B.; GAGNE, G. **Fundamentos de Sistemas Operacionais**. 9. ed. LTC, 2015.
- [2] DEITEL, H.; DEITEL, P. J.; CHOFFNES, D. R. **Sistemas Operacionais**. 3. ed. São Paulo: Pearson, 2005.
- [3] TANENBAUM, A. S.; BOS, H.. **Sistemas Operacionais Modernos**. 4a ed. Pearson, 2016.
- [4] MAZIERO, C. **Sistemas Operacionais: Conceitos e Mecanismos**. Online. Caderno de Exercícios. 2013.