

REVIEW QUESTIONS

6.1 Give examples of reusable and consumable resources.

Exemplos de recursos reutilizáveis são: processadores, memória principal e secundária, canais de E/S, arquivos, dispositivos, etc. Exemplos de recursos consumíveis são: sinais, interrupções, mensagens e informações em buffers de E/S.

6.2 What are the three conditions that must be present for deadlock to be possible?

Três condições devem estar presentes para que um deadlock seja possível: **exclusão mútua**, somente um processo pode utilizar um dado recurso por vez; **manter e esperar**, um processo pode manter recursos alocados enquanto espera pela alocação de outros recursos; **nenhuma preempção**, nenhum recurso pode ser retirado de maneira forçada de um processo que o abortou e que o tenha mantido.

6.3 What are the four conditions that create deadlock?

O deadlock pode existir com as três condições mencionadas no exercício 6.2, porém uma quarta condição se faz necessária: **espera circular**, uma cadeia fechada de processos em estado de espera deve existir, ou seja, cada processo mantém ao menos um recurso que está sendo requisitado pelo próximo na cadeia de processos.

6.4 How can the hold-and-wait condition be prevented?

Por esta condição poderíamos exigir que todos os processos requisitassem todos os recursos ao mesmo tempo, em vez de uma por vez com alguma eventual espera.

6.5 List two ways in which the no-preemption condition can be prevented.

Se um processo requer um recurso que está sendo mantido por um outro processo, o sistema operacional pode preemptar o segundo processo e requerer a liberação dos recursos por parte deste processo. Ou se um processo que está mantendo recursos é recusado, então este processo deve liberar seus recursos e se necessário solicitá-los novamente com a adição de mais recursos.

PROBLEMS

6.2 Show how each of the techniques of prevention, avoidance, and detection can be applied to Figure 6.1.

- a. Q aloca B e depois A, e então libera B e A. Quando P assume a execução, ele irá ser capaz de alocar ambos os recursos.

- b. Q aloca B e depois A. P executa e bloqueia quando solicita A. Q libera B e A. Quando P reassume a execução, ele será capaz de alocar ambos os recursos.
- c. Q aloca B e então P aloca A. O deadlock será inevitável, dado que a execução de ambos irá prosseguir, Q irá bloquear em A e P irá bloquear em B.
- d. P aloca A e então Q aloca B. O deadlock será inevitável, dado que nos prosseguimentos da execução, Q irá bloquear em A e P irá bloquear em B.
- e. P aloca A e depois B. Q executa e bloqueia na requisição de N. P libera A e B. Quando Q reassume a execução, ele irá ser capaz de adquirir ambos os recursos.
- f. P aloca A e depois B e então libera A e B. Quando Q reassume a execução, ele poderá alocar ambos os recursos.

6.5 Given the following state for the Banker's Algorithm. i) 6 processes P0 through P5; ii) 4 resource types: A (15 instances); B (6 instances); iii) C (9 instances); D (10 instances); iv) Snapshot at time T0:

a. Verify that the Available array has been calculated correctly

Available array = Instances – Process(n)

$$A = 15 - (2+0+4+1+1+1) = 6$$

$$B = 6 - (0+1+1+0+1+0) = 3$$

$$C = 9 - (2+1+0+0+0+1) = 5$$

$$D = 10 - (1+1+2+1+0+1) = 4$$

b. Calculate the Need matrix

Need matrix = MaximumDemand – CurrentAllocation

7 5 3 4

2 1 1 2

3 4 4 2

2 3 3 1

4 1 2 1

4 4 3 3

c. Show that the current state is safe, that is, show a safe sequence of processes. In addition, to the sequence show how the Available (working array) changes as each process terminates.

Comparando o Available Array com a Need Matrix começamos pelo processo P1.

Em seguida tem-se $[6,3,5,4] - [2,1,2,2] = [4,2,3,2]$ que é terminado. E então retorna os recursos $[4,2,3,2] + [2,2,3,3] = [6,4,6,5]$.

O processo P2 é feito da seguinte forma $[6,4,6,5] - [3,4,4,2] = [3,0,2,3]$ que é terminado. E então retorna os recursos $[3,0,2,3] + [7,5,4,4] = [10,5,6,7]$.

O processo P3 é feito da seguinte forma $[10,5,6,7] - [2,3,3,1] = [8,2,3,6]$ que é terminado. E então retorna os recursos $[8,2,3,6] + [3,3,3,2] = [11,5,6,8]$.

O processo P4 é feito da seguinte forma $[11,5,6,8] - [4,1,2,1] = [7,4,4,7]$ que é terminado. E então retorna os recursos $[7,4,4,7] + [5,2,2,1] = [12,6,6,8]$.

O processo P5 é feito da seguinte forma $[12,6,6,8] - [3,4,3,3] = [9,2,3,5]$ que é terminado. E então retorna os recursos $[9,2,3,5] + [4,4,4,4] = [13,6,7,9]$.

O processo P0 é feito da seguinte forma $[13,6,7,9] - [7,5,3,4] = [6,1,4,5]$ que é terminado. E então retorna os recursos $[6,1,4,5] + [9,5,5,5] = [15,6,9,10]$.

Com todos os processos são finalizados em segurança a sequência [P0, P1, P2, P3, P4, P5] também é.

d. Given the request (3,2,3,3) from Process P5. Should this request be granted? Why or why not?

$$A = 15 - (2+0+4+1+1+3) = 4$$

$$B = 6 - (0+1+1+0+1+2) = 1$$

$$C = 9 - (2+1+0+0+0+3) = 3$$

$$D = 10 - (1+1+2+1+0+3) = 2$$

A sequência [4,1,3,2] é uma necessidade para P5 que é [1,2,1,1]. Como não é um estado seguro o pedido não será atendido.