

2014

Curso Preparatório de Arquitetura de Computadores e Sistemas Operacionais



Capítulo 7.1 - DEADLOCK

Objetivo do curso: abordar os principais conceitos de arquitetura de Computadores e de Sistemas Operacionais e resolver questões de provas.

Instrutor: Emanuel Peixoto

Atualizado em: 21/08/2014

Conteúdo

7.1. DEADLOCK.....	3
✓ Condições par ocorrência de Deadlock	4
✓ Estratégias para Tratar Deadlocks	4
✓ Grafo de Alocação de Recursos	5
✓ Identificação de um DEADLOCK.....	5
✓ Correção do deadlock.....	6
QUESTÕES DE CONCURSOS	7



7.1. DEADLOCK

Conceitos fundamentais

- **Deadlock** - é uma situação em que um processo aguarda por um recurso que nunca estará disponível ou um evento que não ocorrerá.
- **Starvation ou espera indefinida ou inanição de processo** – É a condição onde um processo nunca consegue ser alocado para execução.
 - Originado por **algoritmo de prioridade** ou de execução **aleatória** de processos.
- **Livelock** – É a situação onde um processo não consegue avançar em por causa da **mudança de estado da outra** (ex. duas pessoas no corredor que sempre escolhem o mesmo lado para desviarem).
 - Pode ocorrer por alteração de desempenho de recursos (CPU, memória), originado por outro processo.
 - Alteração de prioridade durante as execuções
 - O processo **não está bloqueado** por nenhum outro.
 - **É um tipo de inanição ou starvation**
- **Exclusão mútua** – Evita que dois ou mais processos utilizem um recurso simultaneamente
 - Desabilitar Interrupções
 - Variáveis de Impedimento - **TSL (test and set lock)**
 - Alternância obrigatória - **Spin Lock**
- **Semáforo** – É um mecanismo de implementação de **exclusão mútua**
 - É uma variável inteira (Down e UP)
 - É implementado no núcleo do S.O
 - Down decrementa a variável e coloca o processo em espera
 - UP incrementa e desperta o processo
 - **Mutexes** é um **semáforo binário** (down=0, up=1)
- **Monitor** – É um mecanismo de implementação de **exclusão mútua através de software de alto nível**
- Tornam mais simples o desenvolvimento de aplicações concorrentes
- É implementado no **compilador (alto nível)**

Características dos recursos

- **Tipos de Recursos R1, R2, . . . , Rm**
 - ciclos de UCP, espaço de memória, dispositivos de E/S
- Processos precisam acessar recursos ordenadamente
- Supor que um processo mantém um recurso A
 - No mesmo tempo outro processo mantém B e requisita A
 - Os dois são bloqueados e permanecem assim
- **Deadlocks** ocorrem quando ...
 - Processos tem garantido acesso exclusivo aos dispositivos
 - Estes dispositivos são referidos como recursos



<ul style="list-style-type: none">• Recursos preemptáveis<ul style="list-style-type: none">○ Podem ser retirados de um processo sem efeitos negativos○ Exemplo: Memória
<ul style="list-style-type: none">• Recursos não-preemptáveis<ul style="list-style-type: none">○ Causa falha do processo se forem retirados do mesmo○ Exemplo: gravadora de CD
<ul style="list-style-type: none">• Cada processo utiliza um recurso como segue:<ul style="list-style-type: none">○ Requisita○ Usa○ Libera
<ul style="list-style-type: none">• Deve esperar se requisição é rejeitada<ul style="list-style-type: none">○ Processo requisitante pode ser bloqueado○ Pode falhar com código de erro

✓ **Condições par ocorrência de Deadlock**

Deadlock pode existir se 4 condições acontecem simultaneamente

- **Exclusão Mútua**
 - Cada recurso alocado a apenas um processo por vez ou livre
- **Manter e esperar**
 - Processos mantendo recurso podem requisitar recursos adicionais.
- **Sem preempção**
 - Um recurso previamente garantido não pode ser retirado do processo
- **Espera Circular**
 - Deve haver uma cadeia circular de 2 ou mais processos
 - Cada um está esperando por recurso mantido pelo próximo membro da cadeia

✓ **Estratégias para Tratar Deadlocks**

Tratamento do Deadlock

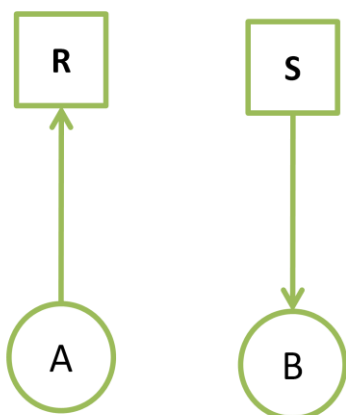
- Assegurar, através da prevenção, que o sistema nunca vai entrar em um estado de deadlock
- Permitir que o sistema entre em estado de deadlock e então, detecte e corrija
- Evitar, através da alocação cuidadosa dos recursos.
- Ignorar o problema e fazer de conta que deadlock nunca ocorrem no sistema



✓ Grafo de Alocação de Recursos

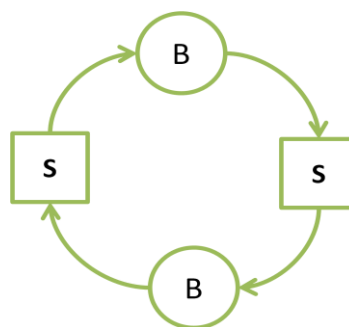
Um conjunto de vértices V e um conjunto de arcos A.
<ul style="list-style-type: none"> V é particionado em 2 tipos: <ul style="list-style-type: none"> $P = \{P_1, P_2, \dots, P_n\}$, conjunto dos processos no sistema. $R = \{R_1, R_2, \dots, R_m\}$, conjunto dos tipos de recursos no sistema.
Arco de requisição – arco direto $P_1 \rightarrow R_j$
Arco de atribuição – arco direto $R_j \rightarrow P$

Notação:



a)

b)

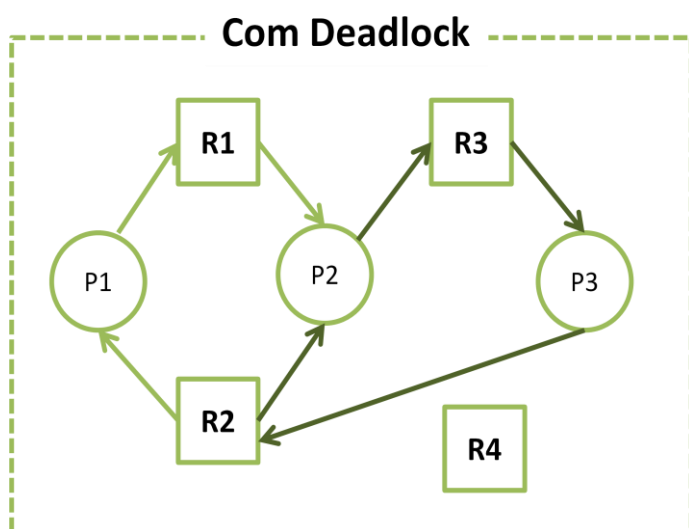
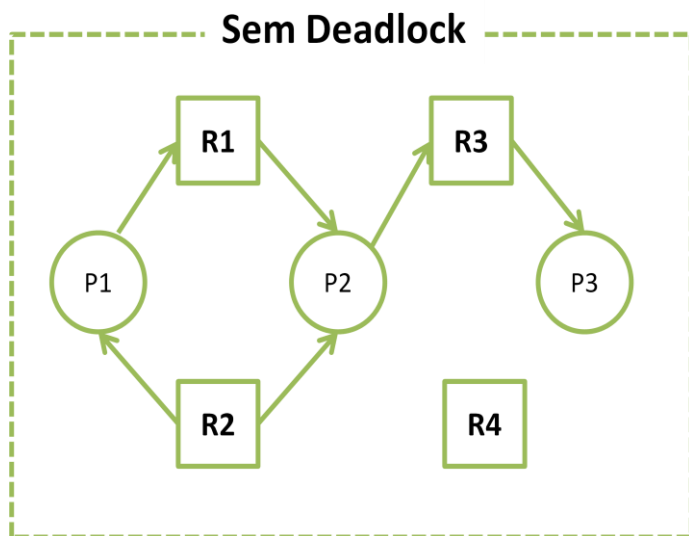


No exemplo a temos o processo “A” requisitando o recurso “R”, no exemplo b, temos o recurso “R” atribuído a “B”

✓ Identificação de um DEADLOCK

Ocorrência
Se o grafo não contém ciclos → Sem deadlock
Se o grafo contém um ciclo → Com deadlock





✓ Correção do deadlock

Correção do Deadlock

- **Desalocar o recurso** para o processo bloqueante
 - Sem o recurso, a condição de deadlock é desfeita

Eliminar um dos processos

Voltar atrás (**Rollback**) - retornar para algum estado seguro, reiniciar o processo para aquele estado. Para isto é necessário algum tipo de **checkpoint** para possibilitar reinício.



QUESTÕES DE CONCURSOS

(EsFCEEx/2011)

53. Uma situação de livelock envolvendo dois processos difere de uma situação de deadlock porque:
- (A) nenhum processo fica bloqueado.
 - (B) os processos não ocupam recursos desnecessários.
 - (C) ambos os processos ficam bloqueados simultaneamente.
 - (D) os processos bloqueiam e liberam recursos indefinidamente.
 - (E) um dos processos segue executando e o outro fica bloqueado indefinidamente.

(Marinha /2006)

- 47) Considere um sistema computacional hipotético em que é concedida a posse de somente um recurso por processo de cada vez. Em relação à ocorrência de DEADLOCK neste sistema, é correto afirmar que
- (A) cabe ao usuário controlar o *DEADLOCK*.
 - (B) embora muito raro, o sistema entra em *DEADLCK*.
 - (C) o sistema é livre de *DEADLOCK*.
 - (D) o sistema sempre vai entrar em *DEADLOCK*.
 - (E) se o sistema entrar em *DEADLOCK*, ele conseguirá desarmá-lo sem intervenção externa.



(Marinha/2008)

- 22) Sobre as soluções para o problema da seção crítica na comunicação interprocessos, que satisfazem à exigência da exclusão mútua, é correto afirmar que
- (A) a desabilitação das interrupções em um ambiente multiprocessado impede que mais de um processo leia e escreva, ao mesmo tempo, na memória compartilhada.
 - (B) a instrução de hardware *test and set lock* (TSL) é caracterizada por não realizar espera ociosa (ou espera em ação).
 - (C) a atomicidade é absolutamente essencial nas duas operações padrões sobre semáforos.
 - (D) *mutexes* são versões simplificadas de semáforos, em que seu valor inteiro pode variar dentro de um domínio irrestrito.
 - (E) monitores são estruturas de baixo nível que implementam exclusão mútua de forma automática.

(Marinha /2008)

- 25) Considere a carga de trabalho apresentada a seguir.

<u>Processo</u>	<u>Tempo de Pico</u>
P ₁	11
P ₂	30
P ₃	4
P ₄	8
P ₅	13

Os processos chegam no tempo 0, na ordem P₁, P₂, P₃, P₄ e P₅, com tempo de duração do pico de CPU dado em milisegundos. Supondo que os cinco processos possuem a mesma prioridade e que um *quantum* equivale a 10 milisegundos, assinale a opção que contém o algoritmo de escalonamento (*scheduling*) de CPU que apresentará o menor tempo de espera.

- (A) First come, first served (FCFS).
- (B) Shortest job first (SJF).
- (C) FIFO.
- (D) Round-Robin.
- (E) Por prioridades.



(Marinha/2008)

27) Analise a seguinte seqüência de alocação de recursos.

- 1- O processo P_1 possui o recurso R_1 e requisita o recurso R_2 ;
- 2- O processo P_2 nada possui e requisita o recurso R_3 ;
- 3- O processo P_3 nada possui e requisita o recurso R_2 ;
- 4- O processo P_4 nada possui e requisita os recursos R_2 , R_3 e R_5 ;
- 5- O processo P_5 possui o recurso R_3 e requisita o recurso R_4 ;
- 6- O processo P_6 possui o recurso R_6 e requisita o recurso R_2 ;
- 7- O processo P_7 possui o recurso R_4 e requisita o recurso R_7 ;
- 8- O processo P_8 possui o recurso R_7 e requisita o recurso R_5 .

Com base nas informações acima, assinale a opção correta.

- (A) Os processos P_1 , P_3 , P_4 e P_6 sofrem *deadlock*.
- (B) Se P_3 requisitar o recurso R_1 , os processos P_1 e P_3 sofrerão *deadlock*.
- (C) Os processos P_4 , P_5 , P_7 e P_8 estão na condição de espera circular.
- (D) Se P_4 obtiver o recurso R_5 , os processos P_4 , P_5 , P_7 e P_8 sofrerão *deadlock*.
- (E) O estado de alocação de recursos atual é inseguro.

(Marinha/2010)

10) Considere um sistema em *deadlock*, que tem quatro processos, de "A a D", e quatro recursos, de "O a R". Os recursos estão sendo usados e requisitados, conforme o quadro abaixo:

1. O processo "A" possui o recurso "O" e requisita o recurso "P".
2. O processo "B" nada possui, mas requisita o recurso "O".
3. O processo "C" possui o recurso "Q" e requisita o recurso "P".
4. O processo "D" possui o recurso "R" e requisita os recursos "Q" e "P".

Nessas condições, assinale a opção que apresenta os processos envolvidos no *DEADLOCK*.

- (A) A e B
- (B) C e D
- (C) B e C
- (D) A e C
- (E) A e D.



(Marinha/2011)

- 11) Na comunicação entre processos, que solução de exclusão mútua tem a característica de precisar de espera ociosa para evitar condições de disputa?
- (A) Instrução TSL (Test and Set Lock).
 - (B) Semáforo.
 - (C) Mutex (Mutual Exclusion).
 - (D) Monitor.
 - (E) Troca de mensagens.

(Marinha/2011)

- 34) Assinale a opção que apresenta uma condição necessária para que ocorra um *deadlock*.
- (A) Reversão de estado.
 - (B) Eliminação de processos.
 - (C) Preempção.
 - (D) Exclusão mútua.
 - (E) Spooling.

(Marinha/2013)

- 3) Uma coleção de rotinas, de variáveis e de estruturas de dados, agrupados em um tipo especial de módulo ou pacote que permite que, em um dado momento, somente um processo ativo execute um de seus procedimentos, implementando, dessa forma, a exclusão mútua, é denominada
- (A) semáforo.
 - (B) mutexes.
 - (C) instrução TEST AND SET LOCK.
 - (D) monitor.
 - (E) BUSY WAITING.



(Marinha/2013)

28) As informações a seguir se referem ao estado de dois processos, P1 e P2, no decorrer do tempo t, para serem executados em uma única CPU:

- em t1 = 0ms, P1 = PRONTO, P2 = PRONTO
- em t2 = 1ms, P1 = EXECUÇÃO, P2 = PRONTO
- em t3 = 3ms, P1 = ESPERA, P2 = EXECUÇÃO
- em t4 = 6ms, P1 = ESPERA, P2 = PRONTO
- em t5 = 8ms, P1 = ESPERA, P2 = EXECUÇÃO
- em t6 = 9ms, P1 = ESPERA, P2 = ESPERA
- em t7 = 11ms, P1 = PRONTO, P2 = ESPERA
- em t8 = 13ms, P1 = EXECUÇÃO, P2 = PRONTO
- em t9 = 15ms, P1 = ENCERRADO, P2 = EXECUÇÃO

Considerando que o algoritmo de escalonamento é do tipo escalonamento preemptivo circular (ROUND-ROBIN), calcule o valor do TIME-SLICE, em ms, do sistema, desprezando os tempos de processamento relativos às funções do Sistema Operacional, e assinale a opção correta.

- (A) 1
- (B) 2
- (C) 3
- (D) 4
- (E) 5

(Marinha/2013)

45) Sabe-se que em um Sistema Operacional (SO) dinâmico, existem vários processos desejando utilizar simultaneamente uma impressora. A estratégia de alocação nesse SO consiste em ceder o uso da impressora ao processo com o menor arquivo a ser impresso e que um dos referidos processos possui um arquivo extenso para imprimir. Tendo em vista que existem diversos outros processos com arquivos menores para utilizar a impressora, pode acontecer de o processo com o arquivo extenso nunca conseguir alocar a impressora, levando este processo à condição de

- (A) DEADLOCK
- (B) SWAPPING
- (C) reentrância
- (D) SPOOLING
- (E) STARVATION



(EsFCEEx/2011)-53-A

(Marinha/2006)-47-C

(Marinha/2008)-22-C

(Marinha/2008)-25-B

(Marinha/2008)-27-D

(Marinha/2010)-10 – Anulada

(Marinha/2011)-11-A

(Marinha/2011)-34-D

(Marinha/2013)-3-D

(Marinha/2013)-28-C

(Marinha/2013)-45-E

