

Deadlocks

Raí Emanuel T. de Freitas

Departamento de Engenharias e Tecnologia - DETEC - Campus Pau dos Ferros.
Universidade Federal Rural do Semi-Árido

10 de fevereiro de 2025

Material em construção

Esse slide está em construção! Novas atualizações ou correções serão aplicadas.

Repositório Github

Repositório Github com materiais da disciplina de sistemas operacionais

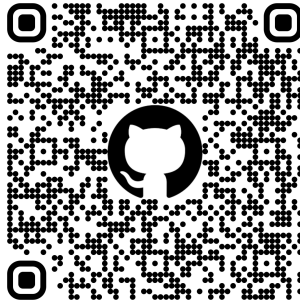


Figura: Caption

Sumário

1 Introdução

2 Deadlock

- Exercícios de fixação
- Caracterização do Deadlock
 - Exercícios de fixação
- Prevenção de Deadlock

Sumário

1 Introdução

2 Deadlock

- Exercícios de fixação
- Caracterização do Deadlock
 - Exercícios de fixação
- Prevenção de Deadlock

Introdução

- Os sistemas multiprogramados surgem na década de 60
 - Partes diferentes dos programas podem ser executadas concorrentemente
 - Aplicação concorrente
- Aplicações concorrentes têm como princípio a execução cooperativa de processos e threads
 - Trabalham em uma mesma tarefa em prol de um resultado comum
- Em sistemas multiprogramados com processador único, os processos alternam sua execução segundo critérios de escalonamento
 - Prioridade, mais recente, menor tempo de execução

Lembre-se

Não há paralelismo real nesse modo exemplificado, mas concorrência de recursos

Introdução

- Os processos compartilham recursos do sistema
 - Arquivos, processador, memória, impressora e entre outros
- Compartilhamento de recursos pode ocasionar situações indesejadas
 - Comprometimento da execução correta das aplicações
- Como evitar esse problema?
 - Processos concorrentes devem ter suas execuções sincronizadas para garantir o correto funcionamento dos programas
 - Gerência feita pelo SO
 - Problema de deadlock

Sumário

1 Introdução

2 Deadlock

- Exercícios de fixação
- Caracterização do Deadlock
 - Exercícios de fixação
- Prevenção de Deadlock

Fonte: [1]

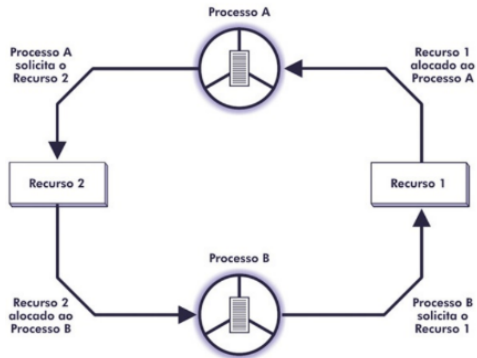


Fig. 7.9 Espera circular.

Sumário

1 Introdução

2 Deadlock

- Exercícios de fixação
- Caracterização do Deadlock
 - Exercícios de fixação
- Prevenção de Deadlock

Exercício de fixação 0

Considere um sistema operacional onde o processo P1 precisa dos recursos R1 e R2 para prosseguir com seu processamento. Concomitantemente, há um processo P2, que também precisa dos recursos R1 e R2. A situação em que, simultaneamente, o processo P1 detém o recurso R1 e espera pelo recurso R2 e o processo P2 detém o recurso R2 e espera pelo recurso R1, é denominada ¹

- (A) Checkout.
- (B) Deadlock.
- (C) Livelock.
- (D) Lockdown.
- (E) Shutdown.

¹fonte: qconcursos.com

Feedback de Resposta Certa - Exercício 0

Resposta certa!

Se você escolheu **B**, essa resposta está correta.

- (B) Deadlock é o problema listado pelo enunciado. A situação ocorre quando dois ou mais processos possuem um recurso e esperam pela liberação do recurso de um outro processo. Forma-se um ciclo de dependência que impede que qualquer um dos processos envolvidos prossiga.

Feedback de Resposta Errada - Exercício 0

Resposta errada!

Se você escolheu **A, C, D ou E**, essa resposta está incorreta.

- (A) Checkout é associado à finalização de compras em sistemas web, por exemplo.
- (C) Livelock é uma condição de mudança constante de estado sem avanço significativo.
- (D) Lockdown não é um termo técnico utilizado em sistemas operacionais normalmente.
- (E) Shutdown é o processo de desligamento do sistema operacional.

Exercício de fixação 1

Acerca dos Sistemas Operacionais, assinale a opção que corresponde à seguinte situação:

Um processo A solicita permissão para usar um recurso X, que lhe é concedido. Um processo B solicita permissão para usar um recurso Y, que também lhe é concedido. E, então, o processo A solicita acesso ao recurso Y, mas a requisição é suspensa até que B o libere. Entretanto, ao invés de liberar o recurso Y, o processo B requisita acesso ao recurso X.²

- (A) Impasse (Deadlock).
- (B) Região Crítica.
- (C) Condição de Corrida.
- (D) Inanição (Starvation).
- (E) Exclusão Mútua.

²fonte: qconcursos.com

Feedback de Resposta Certa - Exercício 1

Resposta certa!

Se você escolheu **A**, essa resposta está correta.

- (A) Um deadlock é causado pela espera infinita por parte de dois processos. Cada um espera a liberação de um recurso necessário para continuar. Cria-se um ciclo de dependência circular típico de um deadlock.

Feedback de Resposta Errada - Exercício 1

Resposta errada!

Se você escolheu **B, C, D ou E**, essa resposta está incorreta.

- (B) Em construção
- (C) Em construção
- (D) Em construção
- (E) Em construção

Exercício de fixação 2

Dois processos, “A” e “B”, querem enviar para a impressora um arquivo que está em um pen-drive, estando esses recursos disponíveis. “A” solicita e consegue acesso à impressora. “B” solicita e consegue acesso ao arquivo. Agora, “A” solicita acesso ao arquivo, mas é negado até que “B” o libere. Porém, “B”, agora, solicita a impressora, que também é negada, até que “A” a libere. Assim, os dois processos são bloqueados. A situação descrita mostra a ocorrência de:³

- (A) Fila.
- (B) Pilha.
- (C) Deadlock.
- (D) Preempção.
- (E) Interrupção.

³fonte: qconcursos.com

Feedback de Resposta Certa - Exercício 2

Resposta certa!

Se você escolheu **C**, essa resposta está correta.

(C) Em construção

Feedback de Resposta Errada - Exercício 2

Resposta errada!

Se você escolheu **A, B, D ou E**, essa resposta está incorreta.

- (A) Em construção
- (B) Em construção
- (D) Em construção
- (E) Em construção

Sumário

1 Introdução

2 Deadlock

- Exercícios de fixação
- **Caracterização do Deadlock**
 - Exercícios de fixação
- Prevenção de Deadlock

Caracterização de deadlocks

Fonte: [2]

System Deadlocks

E. G. COFFMAN, JR.

Pennsylvania State University, University Park, Pennsylvania

M. J. ELPHICK

University of Newcastle upon Tyne, Newcastle upon Tyne, England

A. SHOSHANI

System Development Corporation, Santa Monica, California

A problem of increasing importance in the design of large multiprogramming systems is the, so-called, deadlock or deadly-embrace problem. In this article we survey the work that has been done on the treatment of deadlocks from both the theoretical and practical points of view.

Key words and phrases: deadlocks, deadly embraces, system deadlocks, multiprogramming, interlock problems

CR categories: 4.10, 4.32

Caracterização de deadlock

- Quatro condições simultâneas são necessárias para ocorrência de um deadlock:
 - (1) Exclusão mútua: Um recurso só pode estar alocado a um processo em um determinado instante de tempo
 - (2) Espera por recurso: Um processo pode estar esperando por um recurso que ainda não possui
 - (3) Não preempção: Um recurso não pode ser liberado por um processo só porque outros processos o desejam
 - (4) Espera circular: Um processo pode ter de esperar o recurso de um outro processo, e vice-versa

Caracterização de deadlock

- O deadlock existe em qualquer sistema multiprogramado
 - Técnicas de prevenção, detecção e correção devem levar em conta o tipo do sistema e o impacto no desempenho
 - Deadlock em sistema multiprogramado de tempo real
 - Deadlock em sistema multiprogramado de tempo compartilhado

Sumário

1 Introdução

2 Deadlock

- Exercícios de fixação
- Caracterização do Deadlock
 - Exercícios de fixação
- Prevenção de Deadlock

Exercício de fixação 3

Um impasse pode ser definido formalmente como “um conjunto de processos está em situação de deadlock, se todo processo pertencente ao conjunto estiver esperando por um evento que somente outro processo desse mesmo conjunto poderá fazer acontecer”. Em situações nas quais o deadlock ocorre, nenhum processo consegue executar o recurso que necessita ou liberar o recurso utilizado e nem mesmo ser acordado, pois o recurso necessário está ocupado, causando uma situação de impasse. São condições para entrar no estado de deadlock, EXCETO:⁴

- (A) Exclusão mútua, em que todo recurso está associado a um processo ou disponível.
- (B) Posse e espera, cujos processos que retêm recursos podem solicitar novos recursos.
- (C) Não preempção, cujos recursos concedidos previamente não podem ser forçosamente tomados.
- (D) Espera circular, devendo ser uma cadeia circular de apenas dois processos, na qual cada um está à espera de recursos retidos pelo membro anterior dessa cadeia.

⁴fonte: qconcursos.com

Feedback de Resposta Certa - Exercício 3

Resposta certa!

Se você escolheu **D**, essa resposta está correta.

(D) Em construção

Feedback de Resposta Errada - Exercício 3

Resposta errada!

Se você escolheu **A**, **B**, ou **C**, essa resposta está incorreta.

- (A) Em construção
- (B) Em construção
- (C) Em construção

Sumário

1 Introdução

2 Deadlock

- Exercícios de fixação
- Caracterização do Deadlock
 - Exercícios de fixação
- Prevenção de Deadlock

Prevenção de deadlock

- A prevenção ocorre ao impedir que qualquer uma das quatro condições de existência do deadlock não ocorram
 - Como evitar que essas condições ocorram?

1 - Exclusão mútua: Um recurso só pode estar alocado a um processo em um determinado instante de tempo

Se (1) não ocorrer, então não é necessário esperar o acesso a um recurso, mesmo que já esteja sendo acessado

Prevenção de deadlock

2 - Espera por recurso: Um processo pode estar esperando por um recurso que ainda não possui

- Processos que possuem recursos garantidos não podem solicitar por novos recursos
- Antes do início da execução, o processo deve pré-alocar todos os recursos que vai precisar
- Todos os recursos devem estar disponíveis para o processo começar a execução. Caso contrário, nenhum recurso será alocado e o processo continuará aguardando
- Como consequência pode gerar o desperdício de recursos
 - Recurso alocado por muito tempo e pouca utilização

Prevenção de deadlock

continuação da 2 - Espera por recurso: Um processo pode estar esperando por um recurso que ainda não possui

- Dificuldade em previamente determinar todos os recursos que o processo irá precisar
- Possibilidade de sofrer starvation, caso o conjunto de recursos desejados nunca esteja livre para o processo iniciar a execução

Prevenção de deadlock

3 - Não preempção: Um recurso não pode ser liberado por um processo só porque outros processos o desejam

- Pode ser evitada ao permitir a retirada de um recurso de um processo quando outro processo necessita do mesmo recurso
- Essa liberação forçada pode causar um problema
 - Starvation - O processo com recursos garantidos pode ter que sempre liberá-los para outro processo de maior prioridade. Assim, nunca terminará sua execução

Prevenção de deadlock




4 - Espera circular: Um processo pode ter de esperar o recurso de um outro processo, e vice-versa

- Processos podem apenas manter um recurso por vez
- Alocar novo recurso obriga a liberação do atual
- Limita o compartilhamento de recursos

Prevenção de deadlock

- Prevenir deadlocks evitando qualquer uma das quatro condições de existência é limitador
 - Evita-se utilizar essa abordagem na prática
- É possível evitar o deadlock mesmo que as quatro condições estejam atendidas?
 - Algoritmo do banqueiro (Dijkstra, 1961)
- Algoritmo do banqueiro
 - Os processos informam o número máximo de recursos de cada tipo que serão necessários à sua execução

Referências

-  F. B. Machado and L. P. Maia, *Arquitetura de Sistemas Operacionais*.
Rio de Janeiro, Brasil: Editora LTC, 3 ed., 2005.
Inclui exercícios.
-  E. G. Coffman, M. Elphick, and A. Shoshani, "System deadlocks," *ACM Comput. Surv.*, vol. 3,
p. 67–78, June 1971.
-  A. S. Tanenbaum and H. Bos, *Sistemas Operacionais Modernos*.
Porto Alegre, RS: Bookman, 5 ed., 2024.
E-book. Disponível em: <https://plataforma.bvirtual.com.br>. Acesso em: 09 fev. 2025.

Obrigado!