

Deadlocks

Raí Emanuel T. de Freitas

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

9 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

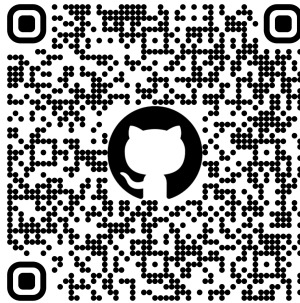


Figura: Caption

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

Deadlock

Fonte: [1]

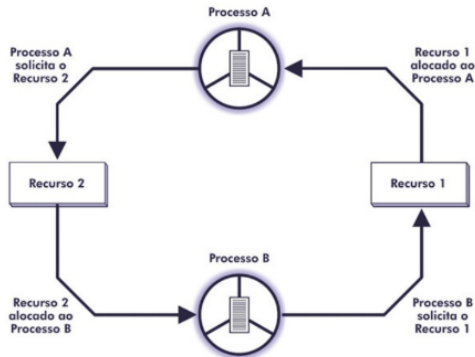


Fig. 7.9 Espera circular.

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) O conceito de deadlock, ou impasse, é um problema comum em sistemas operacionais que trabalham com a alocação de recursos para processos. Ele ocorre quando dois ou mais processos se encontram em um estado onde cada um está esperando por um recurso que está sendo segurado por outro processo, formando assim 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 não é um termo usado em sistemas operacionais para descrever um estado de impasse entre processos.
- (C) Livelock é uma condição onde dois ou mais processos estão efetivamente bloqueados, mas estão constantemente mudando de estado em resposta às mudanças do outro processo, sem fazer progresso real.
- (D) Lockdown não é um termo técnico utilizado em sistemas operacionais para descrever um problema de alocação de recursos entre processos.
- (E) Shutdown refere-se ao processo de desligamento de um sistema operacional e não tem relação direta com um estado de impasse entre processos.

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 ocorre quando dois ou mais processos estão bloqueados permanentemente, cada um esperando que o outro libere um recurso necessário para continuar. Na situação descrita: O processo A já possui o recurso X e agora deseja o recurso Y, que está com o processo B. O processo B possui o recurso Y e agora deseja o recurso X, que está com o processo A. Ambos os processos estão esperando indefinidamente, criando 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) Região Crítica: Este conceito refere-se a uma seção do código que não pode ser executada simultaneamente por mais de um processo. Não se aplica aqui, pois o problema não é a execução simultânea, mas sim a espera por recursos.
- (C) Condição de Corrida: Esta situação ocorre quando o resultado de processos concorrentes depende da ordem não controlada de execução. Não é o caso, já que o problema é a espera indefinida e não a ordem de execução.
- (D) Inanição (Starvation): Refere-se a um processo que nunca obtém os recursos necessários para prosseguir, mas isso é devido à falta de prioridade, não por causa de uma espera circular como no deadlock.
- (E) Exclusão Mútua: Refere-se a uma condição em que os recursos não podem ser compartilhados simultaneamente. Embora a exclusão mútua seja uma das condições que podem levar a um deadlock, por si só, não explica o cenário de

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) O enunciado descreve uma situação onde dois processos, "A" e "B", estão competindo por dois recursos: uma impressora e um arquivo em um pen-drive. Quando "A" obtém acesso à impressora e "B" ao arquivo, ambos entram em um estado onde nenhum pode prosseguir porque cada um está esperando que o outro libere o recurso que possui. Isso caracteriza uma situação de deadlock. Em termos de conceito, o deadlock ocorre quando um conjunto de processos está bloqueado porque cada processo está aguardando um recurso que está sendo ocupado por outro processo do mesmo conjunto. Na questão, isso é exemplificado pelos processos "A" e "B" que estão mutuamente bloqueados.

Feedback de Resposta Errada - Exercício 2

Resposta errada!

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

- (A) fila: Uma fila é uma estrutura de dados onde os elementos são inseridos no final e removidos no início (FIFO: First In, First Out). No contexto de gerenciamento de processos, fila não descreve a situação apresentada, pois não se trata de uma questão de ordem de chegada ou espera em fila.
- (B) pilha: Uma pilha é outra estrutura de dados, mas com comportamento de inserção e remoção no topo (LIFO: Last In, First Out). Novamente, essa estrutura não se aplica ao cenário da questão, que trata de bloqueio de recursos e não de gerenciamento de ordem de processos ou dados.
- (D) preempção: Preempção ocorre quando o sistema operacional interrompe um processo em execução para dar lugar a outro processo. Embora a preempção seja um conceito importante em sistemas operacionais, ela não se relaciona ao problema de deadlock descrito na questão, já que o deadlock é caracterizado pela

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) Um deadlock (ou impasse) ocorre quando um conjunto de processos fica bloqueado de tal forma que nenhum desses processos consegue continuar, pois estão todos esperando por recursos que estão sendo usados por outros processos desse mesmo conjunto. Para que um deadlock ocorra, quatro condições precisam ser atendidas simultaneamente:

1. Exclusão Mútua: Pelo menos um recurso deve estar em modo não compartilhado; ou seja, apenas um processo por vez pode utilizar o recurso.

(Alternativa A)

2. Posse e Espera: Um processo que já está mantendo pelo menos um recurso também pode solicitar novos recursos que estão sendo mantidos por outros processos. (Alternativa B)

3. Não Preempção: Recursos não podem ser forçosamente retirados dos processos

Feedback de Resposta Errada - Exercício 3

Resposta errada!

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

- (A) fila: Uma fila é uma estrutura de dados onde os elementos são inseridos no final e removidos no início (FIFO: First In, First Out). No contexto de gerenciamento de processos, fila não descreve a situação apresentada, pois não se trata de uma questão de ordem de chegada ou espera em fila.
- (B) pilha: Uma pilha é outra estrutura de dados, mas com comportamento de inserção e remoção no topo (LIFO: Last In, First Out). Novamente, essa estrutura não se aplica ao cenário da questão, que trata de bloqueio de recursos e não de gerenciamento de ordem de processos ou dados.
- (C) A alternativa D está incorreta pois a condição de espera circular não se limita a "uma cadeia circular de apenas dois processos". Ela pode envolver mais processos e recursos, formando um ciclo completo onde cada processo aguarda recursos detidos pelo próximo processo na sequência circular.

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!