

10



Fundação Arco-Íris de Araputanga
FCARP
FACULDADE CATÓLICA
RAINHA DA PAZ

Deadlocks (*Impasse*)

Condições, prevenção, detecção e recuperação

Fundamentos de Sistemas Operacionais



Prof. Thiago Dias Negrão | FCARP

Reconhecer

Identificar um deadlock e as quatro condições necessárias para sua ocorrência.

Analisar

Interpretar grafos de alocação e situações de espera por recursos.

Comparar

Distinguir prevenção, evitação, detecção e recuperação.

- **Como dois processos podem ficar parados mesmo quando nenhum deles apresentou erro?**

Deadlock forma uma espera permanente

03

Processo P1

Possui o recurso R1.

Solicita o recurso R2.

Permanece esperando P2 liberar R2.

Processo P2

Possui o recurso R2.

Solicita o recurso R1.

Permanece esperando P1 liberar R1.

Nenhum processo consegue continuar

Recursos entram em disputa

04

Recursos físicos

Impressora
Dispositivo de armazenamento
Canal de comunicação
Memória

Recursos lógicos

Arquivo
Registro de banco de dados
Mutex
Semáforo

Ciclo de uso

Solicitar o recurso
Usar o recurso
Liberar o recurso

O risco aumenta quando o recurso permite apenas um usuário por vez.

Quatro condições precisam ocorrer juntas

05

1. Exclusão mútua

Pelo menos um recurso não pode ser usado simultaneamente.

2. Posse e espera

Um processo mantém um recurso enquanto espera por outro.

3. Não preempção

O sistema não retira o recurso à força. O processo precisa liberá-lo.

4. Espera circular

Cada processo espera um recurso mantido pelo próximo processo do ciclo.

Exclusão mútua limita o compartilhamento

06



Processo P1

Recebeu acesso exclusivo e está imprimindo.

Processo P2

Precisa aguardar a liberação da impressora.

Alguns recursos precisam de exclusividade para funcionar corretamente.

Posse e espera mantém recursos ocupados

07

P1 possui

Arquivo A bloqueado para escrita.



**P1 em espera
o banco de dados**

Outro processo possui

Registro necessário do banco de dados.

Durante a espera, P1 continua impedindo o uso do Arquivo A.

Não preempção impede a retirada forçada

08

Recurso preemptível

O sistema pode interromper o uso e devolver o recurso depois.

Exemplo: tempo de CPU e páginas de memória.

Recurso não preemptível

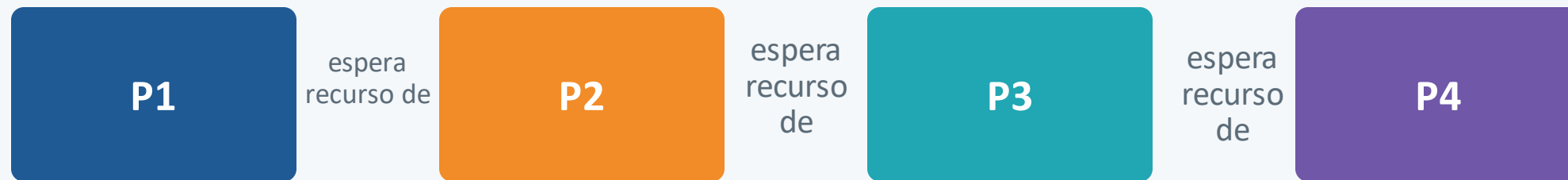
A retirada imediata pode corromper a operação.

Exemplo: arquivo durante gravação ou transação sem rollback.

A possibilidade de preempção depende do tipo de recurso e da operação.

Espera circular fecha o ciclo

09



P4 também espera um recurso mantido por P1: o ciclo se fecha

P1 espera P2, P2 espera P3, P3 espera P4 e P4 espera P1.

Grafo de alocação representa dependências

10



“O grafo representa processos e recursos. Quando o processo aponta para o recurso, significa que está solicitando. Quando o recurso aponta para o processo, significa que está alocado.”

Aresta de solicitação

Do processo para o recurso que ele espera.

Aresta de alocação

Do recurso para o processo que o possui.

Um ciclo exige análise das instâncias

11

Uma instância por recurso

Um ciclo no grafo confirma deadlock.

Cada processo do ciclo espera um recurso que nunca ficará disponível.

Várias instâncias por recurso

Um ciclo indica possibilidade de deadlock.

Outra instância pode ser liberada e permitir o progresso.

Se cada recurso possui somente uma instância, a presença de um ciclo confirma o deadlock. Quando existem várias instâncias, o ciclo mostra um risco, mas outra instância pode ser liberada e permitir a continuação.
Exemplo: uma única impressora é uma instância. Três impressoras representam três instâncias do mesmo tipo de recurso.

Prevenção elimina uma das quatro condições

Na prevenção, o sistema modifica as regras para eliminar uma das quatro condições.

Reduzir exclusão mútua

Compartilhar recursos quando possível. Exemplo: usar spooling para impressão.

Eliminar posse e espera

Solicitar todos os recursos antes de iniciar ou liberar o que já possui.

Permitir preempção

Retirar recursos quando a nova solicitação não puder ser atendida.

Ordenar recursos

Exigir que todos solicitem recursos seguindo uma ordem numérica.

Evitação mantém o sistema em estado seguro

13

Informação necessária

Cada processo informa previamente sua necessidade máxima de recursos.

Teste antes de conceder

O sistema simula a alocação e verifica se ainda existe uma sequência segura.

Decisão

Concede o recurso apenas quando o estado resultante permanece seguro.

Na evitação, o sistema analisa cada solicitação antes de entregar o recurso. Ele verifica se continuará existindo uma ordem segura para todos terminarem.

Detecção aceita o risco e procura o bloqueio

14

1. Permitir

O sistema concede recursos sem impedir antecipadamente todas as combinações perigosas.

2. Examinar

Um algoritmo verifica periodicamente as relações de espera e os recursos disponíveis.

3. Identificar

O sistema localiza os processos que não conseguem concluir e formam o deadlock.

Nessa estratégia, o sistema permite que o deadlock possa acontecer. Depois, executa verificações para descobrir processos que estão presos em uma espera circular. Frequência maior de detecção reduz demora, mas consome processamento.

Grafo de espera detecta ciclos simples

15

P1 espera P2

P2 possui o recurso solicitado por P1.

P2 espera P3

P3 possui o recurso solicitado por P2.

P3 espera P1

P1 possui o recurso solicitado por P3.

Ciclo P1, P2, P3: deadlock detectado

“O grafo de espera liga diretamente os processos. P1 espera P2, P2 espera P3 e P3 espera P1. Como encontramos um ciclo, detectamos um deadlock.”

Recuperação quebra o ciclo

16

“Depois de detectar o deadlock, o sistema precisa quebrar o ciclo.”

Encerrar processos

Finalizar todos os envolvidos ou encerrar um por vez até eliminar o ciclo.

Preemptar recursos

Retirar um recurso de um processo, escolher uma vítima e reiniciar a operação quando possível.

Executar rollback

Retornar o processo a um estado salvo e repetir parte do trabalho.

Qual o sentimento do usuário ao perder o tempo de trabalho?

A escolha considera prioridade, trabalho já realizado e custo da recuperação.

Deadlock, starvation e livelock são diferentes

“Esses problemas parecem semelhantes, mas são diferentes.”

Deadlock

Os envolvidos permanecem bloqueados esperando recursos uns dos outros.

Não há progresso.

Starvation

Uma tarefa pronta espera indefinidamente porque outras recebem preferência.

O sistema continua progredindo.

Livelock

As tarefas continuam reagindo e mudando de estado, mas nenhuma conclui o trabalho.

Exemplo de livelock: duas pessoas tentam desviar uma da outra em um corredor, mas ambas escolhem sempre o mesmo lado.

Estratégias apresentam custos diferentes

Estratégia	Quando age	Vantagem	Limitação	Exemplo
Prevenção	Antes da alocação	Impede deadlock	Reduz uso de recursos	Ordem global
Evitação	A cada pedido	Mantém estado seguro	Exige necessidade máxima	Banqueiro
Deteção	Após ocorrer	Maior flexibilidade	Custo de verificação	Grafo de espera
Recuperação	Após detectar	Restaura progresso	Pode perder trabalho	Rollback

1- P1 está usando a impressora e esperando o scanner.
P2 está usando o scanner e esperando a impressora.
Qual problema está acontecendo?

- a) Escalonamento
- b) Deadlock
- c) Paginação
- d) Fragmentação

2-Quando somente um processo pode utilizar determinado recurso por vez, qual condição está presente?

- a) Exclusão mútua
- b) Memória virtual
- c) Escalonamento
- d) Compartilhamento total

3-Qual ação pode quebrar um deadlock?

- a) Criar mais processos
- b) Aumentar o tempo de espera
- c) Encerrar um dos processos envolvidos
- d) Bloquear mais recursos

Atividade em sala disponível no site.

Síntese da aula

Condições

Deadlock depende das quatro condições ocorrendo ao mesmo tempo.

Tratamento

Prevenção e evitação atuam antes.
Detecção e recuperação atuam depois.

Decisão

Cada estratégia equilibra segurança, desempenho e custo operacional.

Referências básicas

MACHADO, Francis B.; MAIA, Luiz P. Arquitetura de sistemas operacionais. 5. ed. São Paulo: Grupo GEN, 2013.

SILBERSCHATZ, Abraham; GALVIN, Peter B.; GAGNE, Greg. Fundamentos de sistemas operacionais. São Paulo: Grupo GEN, 2015.

Pergunta final: qual estratégia oferece o melhor equilíbrio para um sistema operacional de uso geral?