UniEVANGÉLICA
CENTRO UNIVERSITÁRIO

CURSO DE ENGENHARIA DE SOFTWARE

Disciplina: Sistemas Operacionais

Deadlocks

Prof. M.e Alexandre Tannus

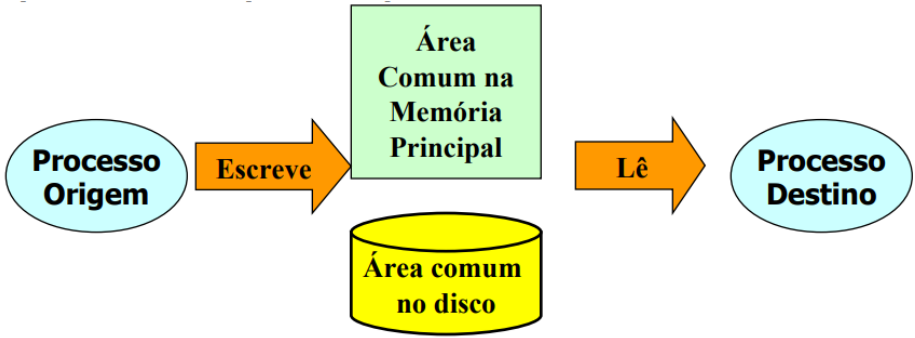
Anápolis - 2021.1
Associação Educativa Evangélica

Questionamentos

- O que fazer se vários processos quiserem o mesmo recurso?
- E se não houver como liberar o recurso?
- Quais estratégias são interessantes para destravar um recurso e evitar a inanição de um processo?

Condições de corrida

- Situações onde dois os mais processos estão lendo ou escrevendo algum dado compartilhado e o resultado depende de quem processa no momento propício.



Formas de Escalonamento

- Não Preemptivo
  - Execução de um processo selecionado até
    - Finalização do processo
    - Bloqueio do processo por E/S
    - Liberação voluntária da CPU pelo processo

- Preemptivo
  - Execução de um processo selecionado por um tempo fixo
  - Suspensão do processo ao final do tempo e seleção de outro processo para execução

- Ivan reserva uma sala para ensinar os colegas, mas não possui marcadores de quadro e cabos para ligar os equipamentos.

- Marta possui marcadores de quadro e cabos para ligar os equipamentos, mas não tem acesso à sala.

Como resolver o problema?

- Dispositivos de *hardware* ou *software* disponibilizados pelo sistema operacional.
- Conjunto finito.
- Apenas um processo pode utilizar o recurso em um dado instante.

- Recursos
  - Marcadores de quadro e cabos
  - Sala de aula

- Processos
  - Ivan
  - Marta

Tipos de recursos

- Preemptivo
  - Pode ser retirado do processo sem efeito prejudicial
    - Memória

- Não preemptivo
  - Não pode ser retirado do proprietário sem causar falhas
    - Impressora
    - Gravador de DVD/CD

Sequência de eventos

- Solicitação do recurso
- Utilização do recurso
- Liberação do recurso

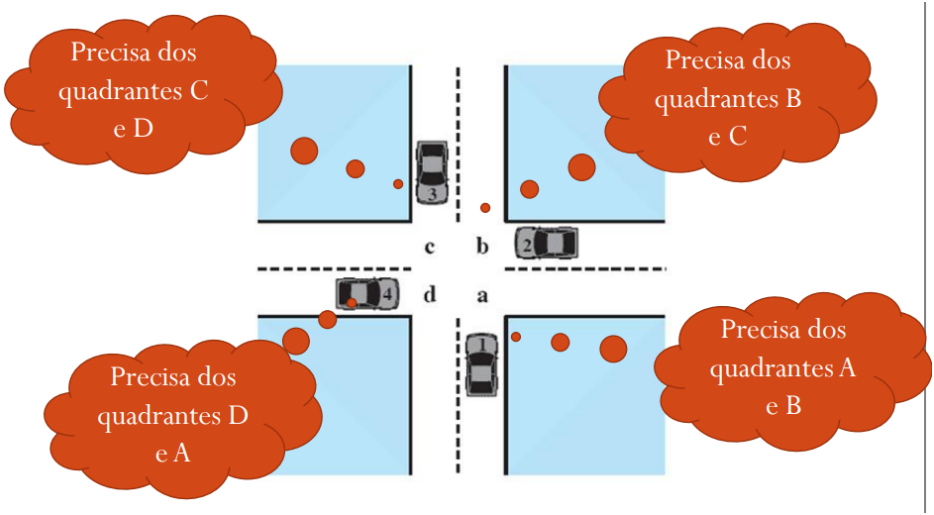
Deadlock

Um conjunto de processos está em um impasse (*deadlock*) se cada processo do conjunto está esperando por um evento que apenas outro processo do conjunto pode causar.

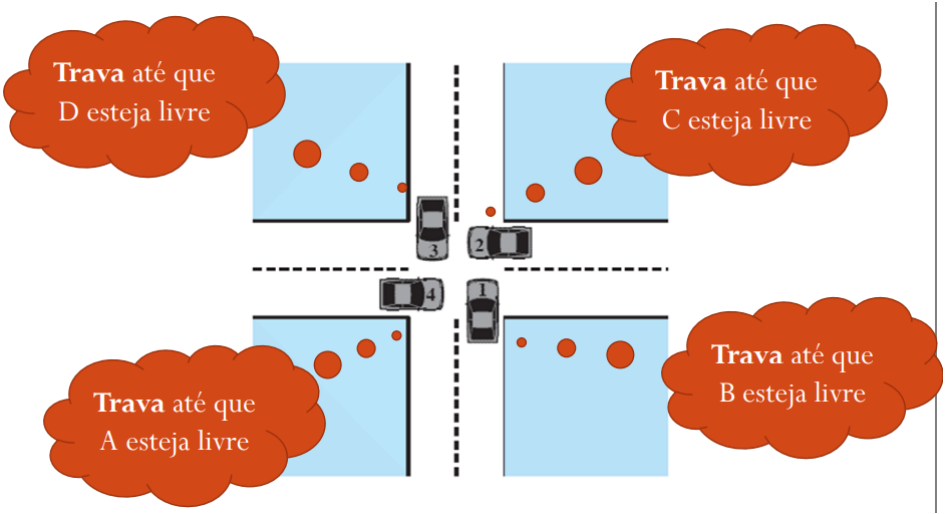
Condições de existência

- Exclusão mútua
  - Um recurso está atribuído a apenas um processo ou disponível para uso
- Posse e espera
  - Processos com recursos alocados podem solicitar novos recursos
- Ausência de preempção
  - Recursos garantidos devem ser liberados voluntariamente pelo processo, sem retirada forçada.
- Espera circular
  - Encadeamento circular entre dois ou mais processos e seus respectivos recursos alocados

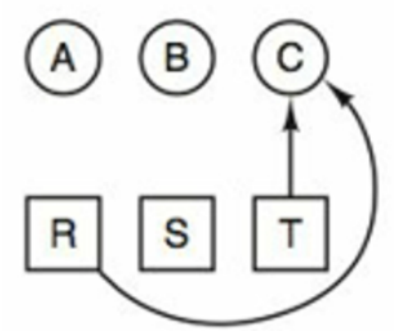
Impasse em potencial

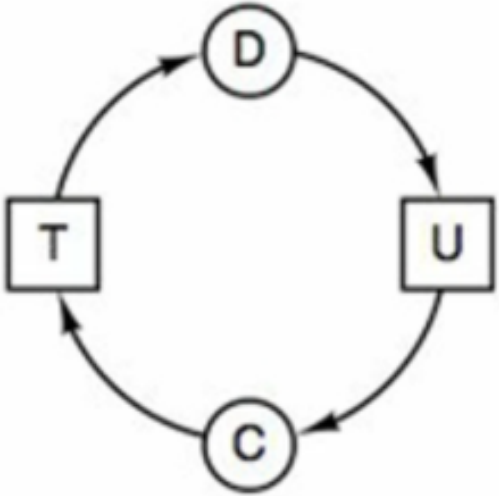


Impasse real



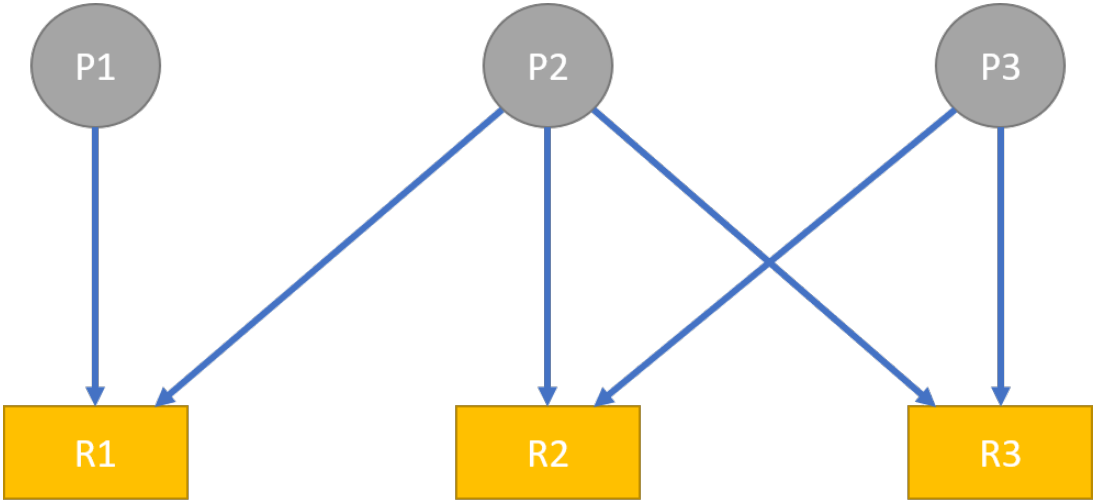
- Grafo de alocação de recursos



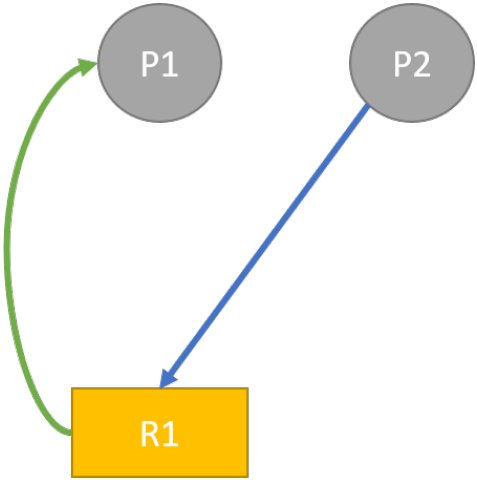

D
T
U
C

Em um sistema computacional multiprocessado, onde o sistema operacional realiza escalonamento de tarefas do tipo preemptivo, três processos (P1, P2 e P3) compartilham recursos (R1, R2 e R3). Os processos P1 e P2 concorrem entre si ao acesso do recurso R1, enquanto P2 e P3 concorrem entre si ao acesso dos recursos R2 e R3. Os recursos R1 e R3 são não preemptíveis e R2 é um recurso preemptível. Todos os três processos usam o mesmo mecanismo de exclusão mútua para garantir acesso exclusivo em suas seções críticas.

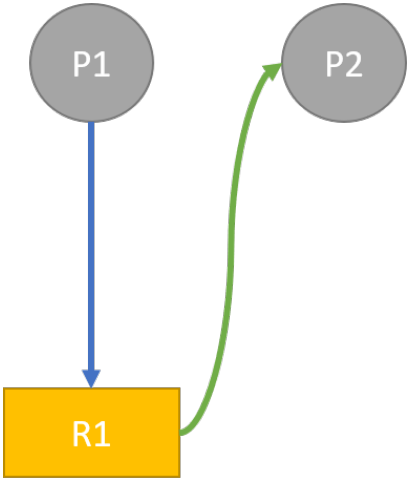
Existe deadlock?



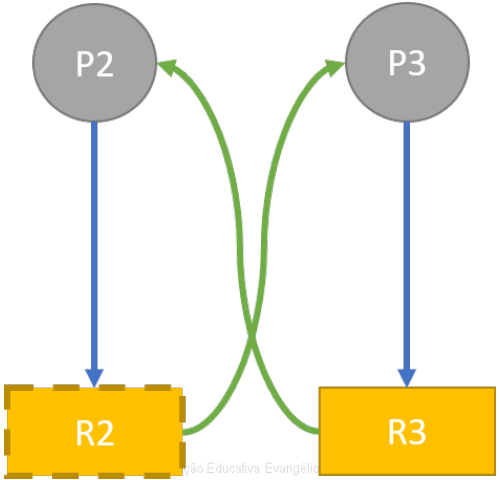
- Recurso R1 é detido pelo processo P1. Processo P1 deseja recurso R1



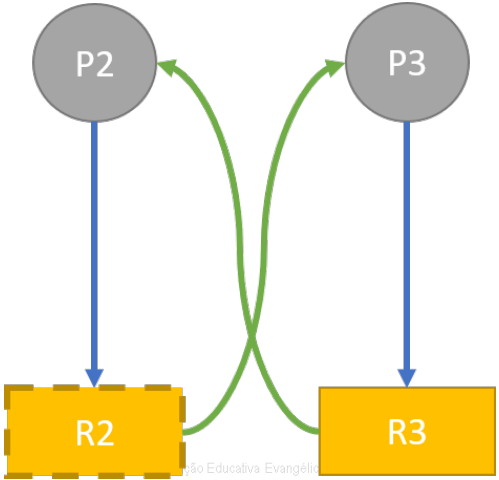
- Recurso R1 é detido pelo processo P2. Processo P1 deseja recurso R1



- Recurso R2 é detido pelo processo P2. Processo P2 deseja recurso R3
- Recurso R3 é detido pelo processo P3. Processo P3 deseja recurso R2

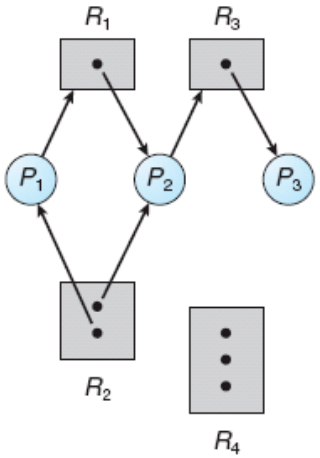


- Recurso R2 é detido pelo processo P3. Processo P2 deseja recurso R2
- Recurso R3 é detido pelo processo P2. Processo P3 deseja recurso R3

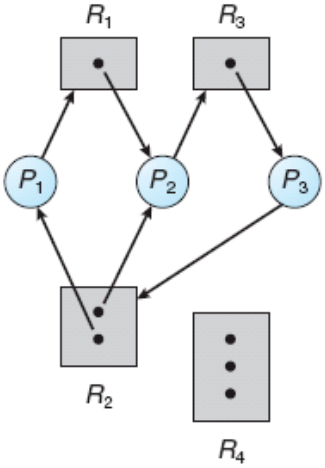


Recursos com múltiplas instâncias

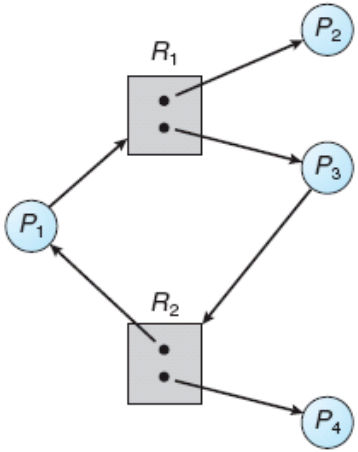
- Alguns recursos podem possuir múltiplas instâncias. Neste caso é necessário avaliar se o ciclo causa *deadlock* ou não



Múltiplas instâncias - Situação 1



Múltiplas instâncias - Situação 2



Estratégias de tratamento

- Ignorar completamente o problema
- Detecção e recuperação
- Evitação dinâmica
- Prevenção por negação estrutural

Algoritmo do Avestruz

- Estratégia mais econômica.
- Ignorar o problema completamente.
- Decisão baseada em
  - Frequência esperada de impasses
  - Frequência que o sistema falha por outros motivos
  - Gravidade do impasse

Detecção e recuperação

- Sistema operacional monitora requisições e liberações de recursos através de um grafo de alocação de recursos
- Em caso de ocorrência de ciclos um processo é finalizado para encerrar o ciclo
- Necessário voltar arquivos para estado original

- Negação de uma condição de existência do *deadlock*
  - Exclusão mútua
  - Posse e espera
  - Ausência de preempção
  - Espera circular

- Necessária para o bom funcionamento de recursos compartilhados
- Impossível ser negada para evitar impasses

- Assegurar que sempre que um processo solicitar um recurso ele não tenha a posse de outro recurso
  - Solicitação de todos os recursos necessários de uma só vez
  - Liberação de recursos retidos antes da solicitação de novos recursos

- Assegurar que sempre que um processo solicitar um recurso ele não tenha a posse de outro recurso
  - Solicitação de todos os recursos necessários de uma só vez
  - Liberação de recursos retidos antes da solicitação de novos recursos

- Problemas
  - Má utilização dos recursos
  - Inanição

Ausência de preempção

- Assegurar que um processo não mantenha a posse de um recurso que não está utilizando (preempção forçada)
  - Recursos liberados entram na lista de solicitações de recursos do processo
  - Facilmente utilizável para recursos como registradores e memória

- Problemas
  - Alguns recursos não podem sofrer preempção forçada (impressoras, *locks*, semáforos)

Estratégias

- Processo pode ter apenas um recurso alocado. Se precisar de um segundo, deve liberar o primeiro
- Numeração global de recursos com solicitação em ordem

Resumo das estratégias

Condição	Estr
Exclusão mútua	Fazer *spool* de tud
Posse e espera	Solicitar todos os r
Ausência de preempção	Permitir preempçã
Espera circular	Ordenar os recurso

Impedimento de deadlock

- Requer informações adicionais sobre a forma de solicitação dos recursos

- O sistema deve considerar
  - Recursos alocados no momento
  - Recursos disponíveis no momento
  - Futuras solicitações e liberações de cada processo

  - Algoritmo do banqueiro para um recurso
  - Algoritmo do banqueiro para múltiplos recursos

- SILBERSCHATZ, A.; GALVIN, P. B.; GAGNE, G.. **Fundamentos de sistemas operacionais: princípios básicos.** Rio de Janeiro: LTC – Livros Técnicos e Científicos, 2013.

- TANENBAUM, A.S., WOODHULL, A.S. **Sistemas Operacionais.** Porto Alegre: Grupo A, 2008.


UniEVANGÉLICA
CENTRO UNIVERSITÁRIO