

**Universidade Federal do Rio Grande do Norte
Unidade Acadêmica Especializada em Ciências Agrárias
Escola Agrícola de Jundiaí
Curso de Análise e Desenvolvimento de Sistemas
TAD0006 - Sistemas Operacionais - Turma 01**

Recursos

Antonino Feitosa
antonino.feitosa@ufrn.br

Macaíba, junho de 2025



Aula Passada

- Sistemas de Paginação
 - Política de Alocação Local e Global
 - Controle de Carga
 - PFF e ultrapaginação
 - Tamanho das Páginas
 - Compartilhamento: páginas e bibliotecas
 - Bibliotecas dinâmicas
- Segmentação
 - Acesso
- Fragmentação Interna e Externa

Roteiro

- Recursos
- Impasses
 - Condições
 - Modelagem
- Detecção e Recuperação de Impasses
- Evitando Impasses
- Prevenindo Impasses
- Livelock e Inanição

Recursos

Recursos

- Recurso: qualquer componente do sistema computacional que pode ser alocados e utilizados por processos.
 - Impressora
 - Dispositivos
 - Arquivos
- Podem ter várias cópias de um mesmo recurso.
 - Três impressoras
- Um recurso é qualquer coisa que precisa ser adquirida, usada e liberada com o passar do tempo.

Recursos

- Em muitas situações os processo precisam se acesso exclusivo a um recurso para completar a sua tarefa.
 - Os processo podem necessitar de acesso exclusivo a vários recursos.
 - Os recursos podem ser de hardware (impressora) ou software (um fragmento de informação, como registro em um banco de dados, por exemplo).

Recursos

- Classificação dos recursos quanto a sua alocação.
 - Recursos preemptíveis
 - Recursos não preemptíveis

Recursos

- Recurso preemptível: é aquele que pode ser retirado do processo proprietário sem causar-lhe prejuízo algum.
 - Memória
 - Depende do contexto: podemos liberar a memória fazendo uso do swapping ou da paginação, no entanto, em sistemas sem esses mecanismos, não podemos liberar memória sem interromper o processo.
- O sistema operacional pode forçar a sua liberação.

Recursos

- Recurso não preemptível: é um recurso que não pode ser tomado do seu proprietário atual sem potencialmente causar uma falha.
 - Scanner
 - O que ocorre se o controle do scanner for passado para outro processo antes dele determinar o processamento atual?
- Precisam ser liberados voluntariamente pelo processo que o detém.

Recursos

- Sequência abstrata de uso de um recurso:
 - Solicitar o recurso
 - Usar o recurso
 - Liberar o recurso
- Se o recurso não estiver disponível, o processo é forçado a esperar.
 - Chamada de sistema.
 - Abertura de arquivos especiais (sistemas UNIX).

Recursos

- Recursos também pode existir no espaço do usuário.
 - O processo precisa gerenciar o uso do recurso.
- Podemos usar semáforos para isso.

FIGURA 6.1 O uso de um semáforo para proteger recursos. (a) Um recurso. (b) Dois recursos.

```
typedef int semaphore;  
semaphore resource_1;
```

```
void process_A(void) {  
    down(&resource_1);  
    use_resource_1( );  
    up(&resource_1);  
}
```

(a)

```
typedef int semaphore;  
semaphore resource_1;  
semaphore resource_2;
```

```
void process_A(void) {  
    down(&resource_1);  
    down(&resource_2);  
    use_both_resources( );  
    up(&resource_2);  
    up(&resource_1);  
}
```

(b)

Recursos

- Na situação b):
- Pode ocorrer que um dos processos adquira ambos os recursos e efetivamente bloqueie o outro processo até concluir seu trabalho.
- Pode acontecer de o processo A adquirir o recurso 1 e o processo B adquirir o recurso 2.
 - Cada um bloqueará agora quando tentar adquirir o outro recurso.
 - Nenhum processo executará novamente.

Impasses



Impasses

Definição: um conjunto de processos estará em situação de impasse (*deadlock*) se cada processo no conjunto estiver esperando por um evento que apenas outro processo no conjunto pode causar.

Impasses: Exemplo

- Suponha, por exemplo, que dois processos queiram cada um gravar um documento escaneado em um disco Blu-ray.
- O processo A solicita permissão para usar o scanner e ela lhe é concedida.
- O processo B é programado diferentemente e solicita o gravador Blu-ray primeiro e ele também lhe é concedido.
- Agora A pede pelo gravador Blu-ray, mas a solicitação é suspensa até que B o libere.
- Infelizmente, em vez de liberar o gravador Blu-ray, B pede pelo scanner.
- A essa altura ambos os processos estão bloqueados e assim permanecerão para sempre.

Impasses

Condições para ocorrência de impasses:

1. Condição de exclusão mútua. Cada recurso está atualmente associado a exatamente um processo ou está disponível.
2. Condição de posse e espera. Processos atualmente de posse de recursos que foram concedidos antes podem solicitar novos recursos.
3. Condição de não preempção. Recursos concedidos antes não podem ser tomados à força de um processo. Eles precisam ser explicitamente liberados pelo processo que os têm.
4. Condição de espera circular. Deve haver uma lista circular de dois ou mais processos, cada um deles esperando por um processo de posse do membro seguinte da cadeia.

Impasses

- Todas as condições listadas devem estar presentes para que um impasse de recurso ocorra.
- Cada condição relaciona-se com uma política que um sistema pode ter ou não:
 - Pode um dado recurso ser designado a mais um processo ao mesmo tempo?
 - Pode um processo ter a posse de um recurso e pedir por outro?
 - Os recursos podem passar por preempção?
 - Esperas circulares podem existir?

Impasses: Modelagem

- Essas condições podem ser modeladas com grafos dirigidos:
- Os grafos têm dois tipos de nós:
 - processos, mostrados como círculos, e recursos, mostrados como quadrados.
- Um arco direcionado de um nó de recurso (quadrado) para um nó de processo (círculo) significa que o recurso foi previamente solicitado, concedido e está atualmente com aquele processo.
- Um arco direcionado de um processo para um recurso significa que o processo está atualmente bloqueado esperando por aquele recurso.

Impasses: Modelagem

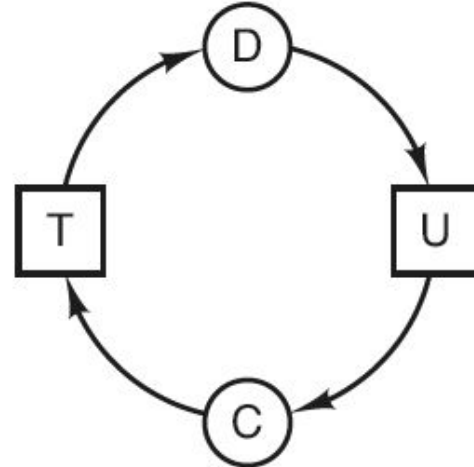
- **FIGURA 6.3** Grafos de alocação de recursos. (a) Processo de posse de um recurso. (b) Solicitação de um recurso. (c) Impasse.



(a)



(b)



(c)

FIGURA 6.4 Um exemplo de como um impasse ocorre e como ele pode ser evitado.

A
Requisita R
Requisita S
Libera R
Libera S

(a)

B
Requisita S
Requisita T
Libera S
Libera T

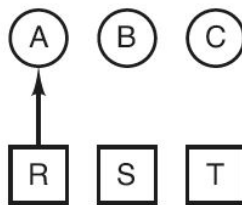
(b)

C
Requisita T
Requisita R
Libera T
Libera R

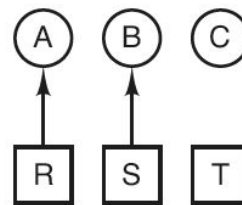
(c)

1. A requisita R
2. B requisita S
3. C requisita T
4. A requisita S
5. B requisita T
6. C requisita R
impasse

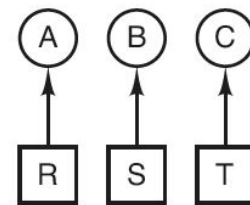
(d)



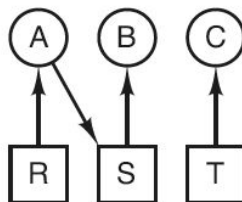
(e)



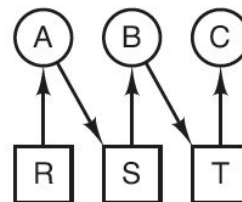
(f)



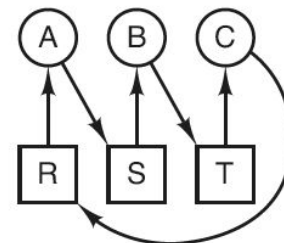
(g)



(h)



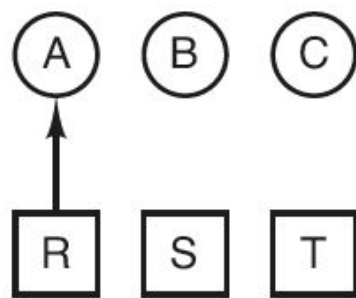
(i)



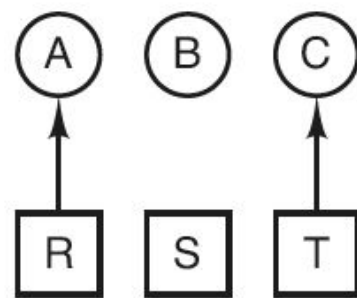
(j)

1. A requisita R
 2. C requisita T
 3. A requisita S
 4. C requisita R
 5. A libera R
 6. A libera S
- nenhum impasse

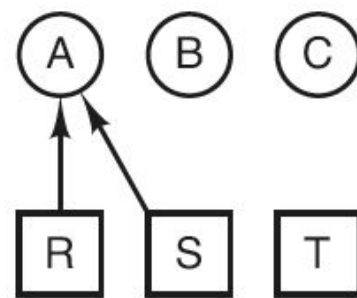
(k)



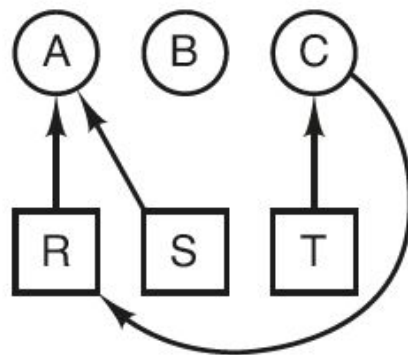
(l)



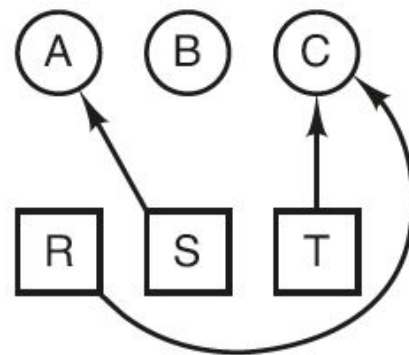
(m)



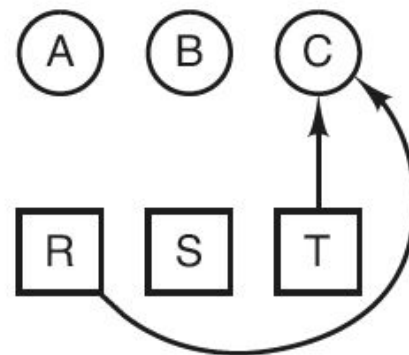
(n)



(o)



(p)



(q)

Impasses: Modelagem

- O impasse pode ocorrer dependendo da ordem de execução.
- Podemos usar a modelagem por grafos para identificar situações que podem levar a um impasse.
 - A formação de um ciclo indica a situação de impasse.
 - O SO pode negar uma solicitação até que seja seguro.

Impasses: Tratamento

- Simplesmente ignorar o problema.
- Detecção e recuperação. Deixe-os ocorrer, detecte-os e tome as medidas cabíveis.
- Evitar dinamicamente pela alocação cuidadosa de recursos.
- Prevenção, ao negar estruturalmente uma das quatro condições.

Algoritmo do Avestruz

Algoritmo do Avestruz

- Algoritmo do avestruz: enfie a cabeça na areia e finja que não há um problema.
- Qual a frequência que se espera que o problema ocorra?
- Qual a frequência com que ocorrem quedas no sistema por outras razões e quão sério um impasse é realmente?
- Devemos considerar o custo de desempenho ou conveniência para eliminar os impasses.

Leitura Complementar

Leitura Complementar

- Capítulo 6 do livro de Sistemas Operacionais, 4ª ed. Andrew Tanenbaum.
 - Detecção e Prevenção de Impasses
 - Evitando Impasses
 - Prevenção de Impasses
 - Livelock
 - Inanição

Resumo

Resumo

- Recursos: Preemptíveis x Não preemptíveis
- Impasses
- Algoritmo do Avestruz
- Detecção e Prevenção de Impasses
- Evitando Impasses
- Prevenção de Impasses
- Livelock
- Inanição

Dúvidas?

