

# Sistemas Operacionais

## Prof. Robson de Souza

### Aulas 21 e 22

**Conteúdo:** Alocação de recursos, impasses e algoritmos

Os S.Os devem garantir o acesso temporário exclusivo de um processo a certos recursos, pois a tentativa de dois processos usarem a mesma entrada na tabela do sistema de arquivos conduzirá a um sistema de arquivos corrompidos.

Recursos → São os objetos acessados (dispositivos, arquivos, etc.). Para alguns recursos, várias instâncias idênticas podem estar disponíveis, se for esse o caso, qualquer uma delas poderá ser utilizada para satisfazer qualquer requisição daquele recurso. Recurso é algo que pode ser adquirido, usado e liberado com o passar do tempo.

Impasse → Também conhecido como deadlock, o impasse ocorre quando um processo A espera um recurso que está sendo usado por um processo B e o processo B espera o recurso que está sendo usado pelo processo A. Os impasses ocorrem quando mais de um processo tem acesso exclusivo a alguns recursos e os impasses podem ocorrer tanto em recursos de hardware quanto em recursos de software. Por exemplo, imagine que um processo de impressão e um processo de scanner requisitaram e obtiveram acesso a uma impressora multifuncional ao mesmo tempo, nesse caso, tem-se um impasse.

### Recursos preemptíveis e não preemptíveis

Um **recurso preemptível** é aquele que pode ser retirado do processo proprietário sem nenhum prejuízo. A memória é um exemplo de recurso preemptível.

Um **recurso não preemptível** é aquele que não pode ser retirado do processo proprietário sem que a computação apresente falha. Um gravador de CD é um exemplo de recurso não preemptível.

Normalmente os impasses envolvem recursos não preemptíveis, pois recursos preemptíveis podem ser retirados e realocados.

### Sequência de eventos necessários ao uso de um determinado recurso:

- 1 – Requisitar o recurso
- 2 – Usar o recurso
- 3 – Liberar o recurso

O processo solicitante será forçado a esperar caso o recurso não esteja disponível. Em alguns S.Os o processo será bloqueado, mas acordado quando o recurso estiver disponível. Em outros sistemas, a falha da requisição gerará um código de erro, o processo deve esperar e tentar novamente.

O processo normalmente vai permanecer em um pequeno laço solicitando o recurso, depois dormirá e tentará novamente. Embora esse processo não esteja bloqueado, será como se estivesse, pois ele não poderá realizar qualquer trabalho útil.

A forma de solicitação de um recurso depende do sistema. Pode permitir uma chamada do tipo request ou abrir arquivos especiais que somente um processo por vez pode abrir usando open. Se o arquivo estiver em uso, o processo chamador será bloqueado até ser fechado pelo proprietário atual.

### Aquisição de recursos

Para alguns tipos de recursos, cabe aos processos de usuário gerenciar o seu uso. Uma maneira de gerenciar os recursos é associar um semáforo a cada recurso. Os semáforos são inicializados com 1, variáveis mutex também podem ser usados. Nesse caso, o Down no semáforo serve para a requisição e utilização do recurso,

Up serve para liberar o recurso.

Um processo pode precisar de mais de um recurso, nesse caso, eles serão adquiridos um após o outro. Quando mais de um processo tenta usar o mesmo recurso, pode existir impasse. A ordem como os recursos são adquiridos também pode gerar impasse.

## Impasses

Um conjunto de processos estará em situação de impasse se todo processo pertencente ao conjunto estiver esperando por um evento que somente outro processo desse mesmo conjunto poderá fazer acontecer. Um processo em situação de impasse não pode ser acordado por, por exemplo, um alarme, pois isso pode causar eventos que liberam outros processos, tudo isso supondo que todos os processos possuem um único thread.

O problema é que cada membro do conjunto de processos em situação de impasse está a espera de um recurso que está sendo usado por outro processo também em impasse. Nenhum deles pode liberar o recurso e nem ser acordado. Não importa o número de processos e, os recursos podem ser de hardware ou software. Esse tipo de impasse é chamado de impasse de recurso.



<https://www.oficinadanet.com.br/post/12786-sistemas-operacionais-o-que-e-deadlock>

## Condições para ocorrência de impasses

Para que um impasse ocorra, quatro condições são necessárias:

- 1 – Exclusão mútua → Cada recurso está em uma de duas situações: Associado a um **único processo** ou disponível.
- 2 – Posse e espera → Processos que retêm recursos concedidos anteriormente e podem requisitar novos recursos.
- 3 – Condição de não preempção → Recursos concedidos a um processo não podem ser retirados de modo forçado. Devem ser liberados pelo processo que os retém.
- 4 – Espera circular → Encadeamento circular de dois ou mais processos, cada um espera um recurso que está sendo usado pelo membro seguinte.

**Todas** essas 4 condições devem estar presentes para que ocorra um impasse, se uma delas faltar, o impasse não ocorrerá. Cada condição está relacionada a uma política que o sistema pode ou não adotar.

## Modelagem de impasses

Os impasses podem ser modelados com grafos onde existem dois tipos de nós:

Círculos → representando processos

Quadrados → representando recursos

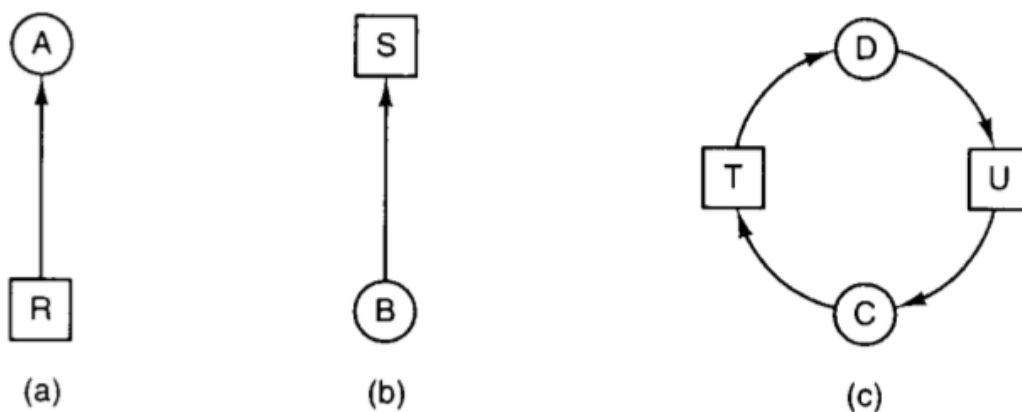
Um arco de um recurso para um processo indica que o recurso foi previamente requisitado, alocado e está sendo usado pelo processo.

Um arco do processo para o recurso indica que o processo está bloqueado esperando pelo recurso. Um ciclo indica um impasse, presumindo que existe um recurso de cada tipo.

Os grafos permitem ver se o impasse vai ocorrer, dessa forma, o S.O pode tomar providências.

Basta modelar as requisições e liberações passo a passo e após cada passo observar o grafo e verificar se ele tem algum ciclo. Se não tiver nenhum ciclo, não haverá impasse. Obviamente que isso vale quando se tem apenas um recurso de cada tipo, mas os grafos de recursos podem ser generalizados para tratar vários recursos do mesmo tipo.

A figura abaixo mostra como é a modelagem de impasses utilizando grafos.



Grafos de alocação de recursos. (a) Segurando um recurso. (b) Solicitando um recurso. (c) Impasse.

Em geral existem quatro estratégias para lidar com impasses:

- 1 – Ignorar por completo o problema.
- 2 – Detecção e recuperação (deixar os impasses ocorrerem, detectá-los e agir).
- 3 – Anulação dinâmica por meio de uma alocação cuidadosa de recursos.
- 4 – Prevenção, negando estruturalmente uma das quatro condições necessárias para gerar um impasse.

## Algoritmo do avestruz

Dependendo do caso, essa solução pode ser razoável ou não. Matemáticos podem achar inaceitável ocorrer impasses. Por outro lado, se o impasse ocorre uma vez a cada 5 anos, mas ocorrem falhas no sistema semanalmente por problemas de hardware, erros de compilação ou problema no S.O, pode não ser vantajoso

perder desempenho para eliminar impasses. Ou seja, depende de até que ponto um impasse pode ser sério.



<http://gustavoinfol.blogspot.com/2016/01/por-que-o-avestruz-enfia-cabeca-no.html>

#### **Referências bibliográficas:**

TANENBAUM, Andrew. 2ª ed. **Sistemas Operacionais Modernos**, Editora Pearson, 2003.

SILBERSCHATZ, Abraham. **Sistemas Operacionais com JAVA**, 6ª ed. Editora Campus

MACHADO, Francis B. **Arquitetura de Sistemas Operacionais**, 4ª ed, LTC, 2007.