

Recursos

➤ Um **recurso** é ou um dispositivo físico (dedicado) do hardware, ou um conjunto de informações, que deve ser exclusivamente usado.



A impressora é um recurso, pois é um dispositivo dedicado, devido ao fato de somente um processo poder usá-la em um dado intervalo de tempo.

- Um processo pode solicitar vários recursos, inclusive várias cópias do mesmo recurso, e pode usar qualquer cópia de um recurso.
- Quando desejar usar um recurso, um processo deverá:
 - **Solicitar o recurso:** esperar pelo recurso, até obtê-lo.
 - **Usar o recurso:** fazer o que for necessário com o recurso.
 - **Liberar o recurso:** devolver o controle do recurso ao sistema.

Recursos

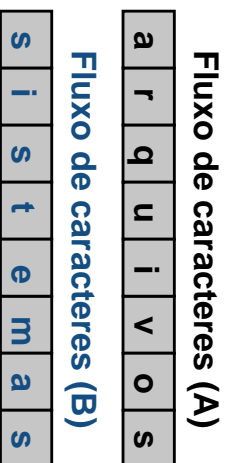
⇨ Existem dois tipos de recursos no sistema:

- ⇨ **Preemptíveis:** pode ser tirado do processo que o possui sem prejudicar o resultado da computação.



A memória é um exemplo de um recurso preemptivo. No exemplo, o processo A está usando a memória.

- ⇨ **Não-preemptíveis:** se o recurso for tirado do processo antes de este liberá-lo, o resultado da computação será incorreto.



A impressora é um dispositivo dedicado, e por isso, é um recurso não-preemptivo. Se os processos A e B imprimem as suas saídas em uma folha, então se A imprimir a saída, liberar a impressora, e B imprimir a sua saída, o resultado será o esperado.

Impasses

- Um conjunto de processos está em um estado de **impasse** se:
- ▢ Cada um dos processos está esperando por um evento que pode ser gerado somente por um outro processo do conjunto.

Processo A



Processo B



Se o processo A possui a impressora, e o processo B possui o CDROM, e se A desejar usar o CDROM, e B desejar usar a impressora, sem liberarem os recursos que possuem, então estes processos esperarão eternamente pelos recursos.

- A tarefa executada pelos processos cooperativos nunca será completada, pois todos os processos estarão esperando.

- Os impasses em geral envolvem recursos não-preemptivos, pois:
- ▢ Se um recurso é preemptivo, os impasses podem ser evitados através da realocação dos recursos aos processos.
 - ▢ Isso não pode ser feito com os recursos não-preemptivos, pois se não o resultado da tarefa cooperativa seria incorreto.

Impasses



Um impasse ocorrerá somente se existirem as seguintes condições:

- ▢ **Condição de exclusão mútua:** cada recurso está atribuído a um único processo em um dado intervalo de tempo.
- ▢ **Condição de segura e espera:** um processo pode solicitar novos recursos quando ainda está segurando outros recursos.
- ▢ **Condição de nenhuma preempção:** um recurso concedido a um processo somente pode ser liberado pelo processo.
- ▢ **Condição de espera circular:** existe uma cadeia circular de dependência entre os processos.



Processo A



Processo B



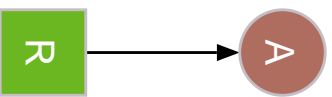
Processo C

No exemplo, os processos A, B, e C possuem, respectivamente, a impressora, o CDROM, e a unidade de fita. Teremos uma dependência circular se A solicitar o uso do CDROM, B o uso da unidade de fita, e C o uso da impressora, sem liberarem os recursos que possuem.

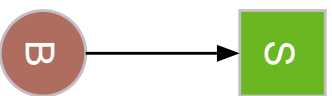
Modelamento dos impasses

⇨ Podemos usar o seguinte **grafo de recursos**, para modelar as solicitações e liberações dos recursos pelos processos.

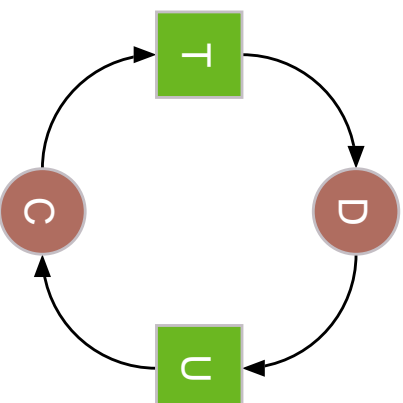
- ⇨ Existe um nó para cada um dos processos (os círculos).
- ⇨ Existe um nó para cada um dos recursos (os quadrados).
- ⇨ Se um recurso está alocado a um processo, existe um arco do nó deste recurso para o nó deste processo.
- ⇨ Se um processo fez uma solicitação a um recurso, existirá um arco do nó deste processo para o nó deste recurso.



O processo A
possui o recurso R



O processo B
deseja o recurso S



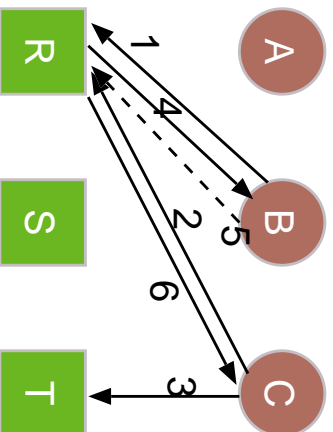
Neste caso, os processos C e D, possuem, respectivamente, os recursos U e T. Existe um estado de impasse porque C solicitou o recurso T, e D o recurso U, sem liberarem os recursos que possuem.

Um estado de impasse,
representado por um ciclo.

Modelagem dos impasses

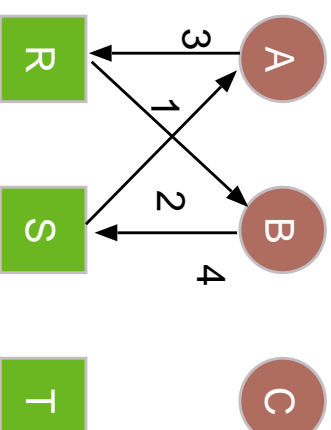
➡ Vamos ver um exemplo prático da utilidade do grafo de recursos, para três processos A, B, e C, que usam três recursos R, S, e T.

➡ Dependendo da ordem em que as solicitações e liberações forem feitas, poderemos ter ou não um impasse:



Exemplo sem impasse:

1. B deseja R
2. C deseja R
3. C deseja T
4. B consegue R
5. B liberta R
6. C consegue R



Exemplo com impasse:

1. R está alocado a B
2. S está alocado a A
3. A deseja R
4. B deseja S

Modelagem dos impasses

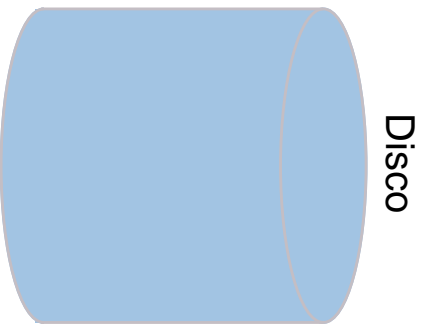
- O grafo que modela o compartilhamento dos recursos:
 - Pode ser usado para detectar se uma dada seqüência de solicitações e de liberações gera ou não um impasse.
 - Pode ser usado pelo sistema operacional para tentar evitar a ocorrência dos impasses.
 - Pode ser facilmente estendido para vários recursos de um mesmo tipo.
- Podemos usar quatro estratégias para tratar dos impasses:
 - Ignorar totalmente a existência dos impasses.
 - Detectar o impasse e recuperar o sistema após a ocorrência deste impasse.
 - Evitar a ocorrência dos impasses em tempo de execução, ao alocar os recursos aos processos.
 - Impedir ocorrência de impasses, definindo regras que impedem a existência de uma das quatro condições necessárias.

Deteccção e recuperação

- ➡ O sistema usa o grafo de recursos para detectar os impasses e recuperar o sistema deste erro:
 - ▢ O sistema atualiza o grafo a cada solicitação ou liberação de um recurso.
 - ▢ Se o sistema detectar um ciclo no grafo:
 - ▢ Um processo do ciclo, escolhido aleatoriamente, é terminado, e os seus recursos são liberados.
 - ▢ A escolha continua até que o grafo seja acíclico.
 - ▢ O problema é que nem sempre é possível reiniciar um processo.
 - ▢ Um outro problema é o de reverter todas as alterações feitas durante a execução de cada um dos processos terminados.
- ➡ Uma outra idéia, mais simples, é a de eliminar um processo que esteja bloqueado por um longo período de tempo.
- ➡ Esta estratégia é em geral usada nos sistemas de lote dos computadores de grande porte.

Prevenção de impasses

- ➡ A idéia é evitar a ocorrência dos impasses garantindo que uma das quatro condições necessárias nunca ocorram.
- ➡ Para evitar a condição da exclusão mútua:
 - ▢ Nenhum dos recursos é dedicado.
 - ▢ Como alguns recursos devem ser dedicados, deveremos definir um outro critério para garantir o uso exclusivo destes recursos.



Disco



Processo spooler

Para alguns recursos que são dedicados, como a impressora, podemos definir que somente um processo, o spooler, solicita a impressora. Para um processo imprimir a sua saída, este deverá criar um arquivo com esta saída, e colocá-lo em um diretório especial, o diretório de spooler.

■ Espaço disponível

Prevenção de impasses



Para evitar a condição de segura e espera:

- ▢ O processo deve solicitar todos os recursos, e começar a sua execução somente após de obter estes recursos, mas:
- ▢ Nem sempre é possível saber todos os recursos necessários.
- ▢ Não conseguiremos otimizar o uso dos recursos:



Se um processo conseguir obter a impressora e o CDROM, e, antes de imprimir, lê algumas informações do CDROM, e leva várias horas para processá-las, a impressora ficará ociosa durante estas horas.

- ▢ O processo também poderia, ao solicitar um recurso:
 - ▢ Liberar todos os recursos que possui.
 - ▢ Se a solicitação puder ser atendida, o processo recebe de volta os recursos que possuía, e o recurso que solicitou.
 - ▢ Caso contrário, o processo perde os recursos que possuía, e espera até conseguir obter todos os recursos necessários.

Prevenção de impasses

➔ Para evitar a condição de nenhuma preempção:

- ▢ O sistema poderá tirar um recurso do processo, se necessário.
- ▢ O problema é que alguns recursos devem ser não-preemptivos, para evitar erros de computação.



Fluxo de caracteres (A)

a r q u i v o s

Fluxo de caracteres (B)

s i s t e m a s

Folha

arquisistemas
vos

Esta idéia claramente não é boa, pois vimos anteriormente que, no caso da impressora, as saídas dos processos seriam intercaladas.



A condição de espera circular pode ser evitada se um processo somente puder usar um recurso em um dado intervalo de tempo:



Suponha que um processo deseja imprimir um arquivo bem grande da unidade de fita. Se o processo não puder usar estes dispositivos ao mesmo tempo, além de a impressão ser incorreta, perderemos um bom tempo para reposicionar a fita na próxima posição do arquivo a ser lido.

Prevenção de impasses

➡ Um outro modo de evitar a condição de espera circular é o de numerar os recursos do sistema:

- ➡ Os processos devem solicitar os recursos em ordem crescente.
- ➡ Neste caso, o grafo de recursos não poderá ter ciclos.
- ➡ Uma alternativa menos restritiva seria a seguinte:
 - ➡ O processo somente pode solicitar um recurso cujo número é maior do que os números de todos os seus recursos.

Recursos numerados:

1. CDROM
2. Impressora
3. Plotadora
4. Unidade de fita
5. Braço autômato



Os recursos serão numerados em ordem crescente. Nesta idéia, uma solicitação de um processo somente será aceita se o número do recurso for maior do que o de todos os recursos que este processo possui.