

Algoritmos de detecção de Deadlock

Prof. Aurélio Hoppe

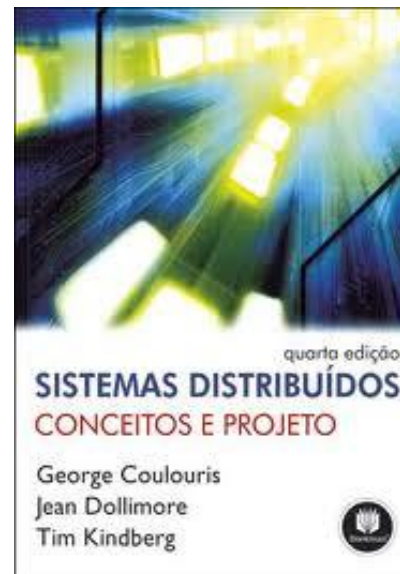
aurelio@furb.br

<http://www.inf.furb.br/~aurelio/>

Grupo de Processamento de Imagens,
Análise de dados, Robótica e
Simulação computacional

. . . aula de hoje

- Como evitar deadlock



Deadlock

Um conjunto de processos estão em situação de Deadlock quando um processo está esperando por um evento que somente outro processo poderá realizar.

Deadlock

Condição para ocorrer deadlock

- O processo “A” requisita e recebe permissão para usar a impressora
- O processo “B” requisita e recebe permissão para usar o DVD
- O processo “A” pede para usar o DVD, mas sua requisição é negada até que “B” libere o recurso
- O processo “B” pede para usar a impressora, mas sua requisição é negada até que “A” libere o recurso

Deadlock

Como tratar a ocorrência de deadlock:

- Ignorar
- Prevenir
- Detectar

Deadlock

Como tratar a ocorrência de deadlock

- Ignorar
 - Algoritmo avestruz
 - Ignora o problema – não possui mecanismo de tratamento
 - É a solução adotada pela maior parte dos sistemas tradicionais (UNIX, Windows)
 - Usuários detectam um problema e começam a matar os processos problemáticos manualmente até que o sistema volte a funcionar

Deadlock

Como tratar a ocorrência de deadlock

- Prevenção
 - Tornam impossível a ocorrência de deadlock. Modela o sistema para que isso não aconteça
 - Os processos só podem deter posse de um único recurso por vez
 - Os processos pedem todos os recursos antecipadamente

Deadlock

Como tratar a ocorrência de deadlock

- Detecção
 - Num SO convencional quando um deadlock ocorre, para resolve-lo matasse um ou mais processos
 - Na detecção primeiro espera acontecer e depois trata-o
 - Existem 2 algoritmos para tratar deadlock:
 - centralizado
 - distribuído

Algoritmo centralizado

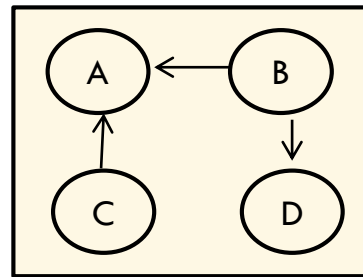
- Cada máquina possui um grafo para seus próprios processos e recursos



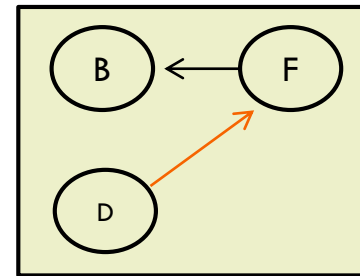
- Um coordenador central detém o grafo de recursos para todo o sistema. Quando o coordenador detecta um deadlock ele mata um dos processos envolvidos

Manutenção da informação

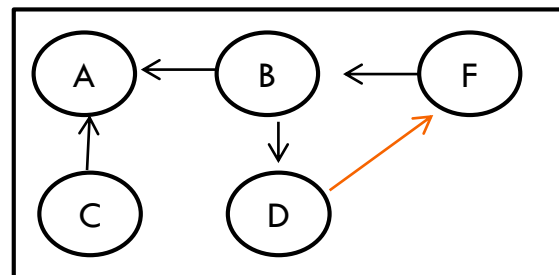
- Sempre que uma aresta for adicionada ou removida o coordenador é informado
- Cada máquina em tempos em tempos envia as informações para o coordenador
- O coordenador solicita as informações quando forem necessárias



Maquina A

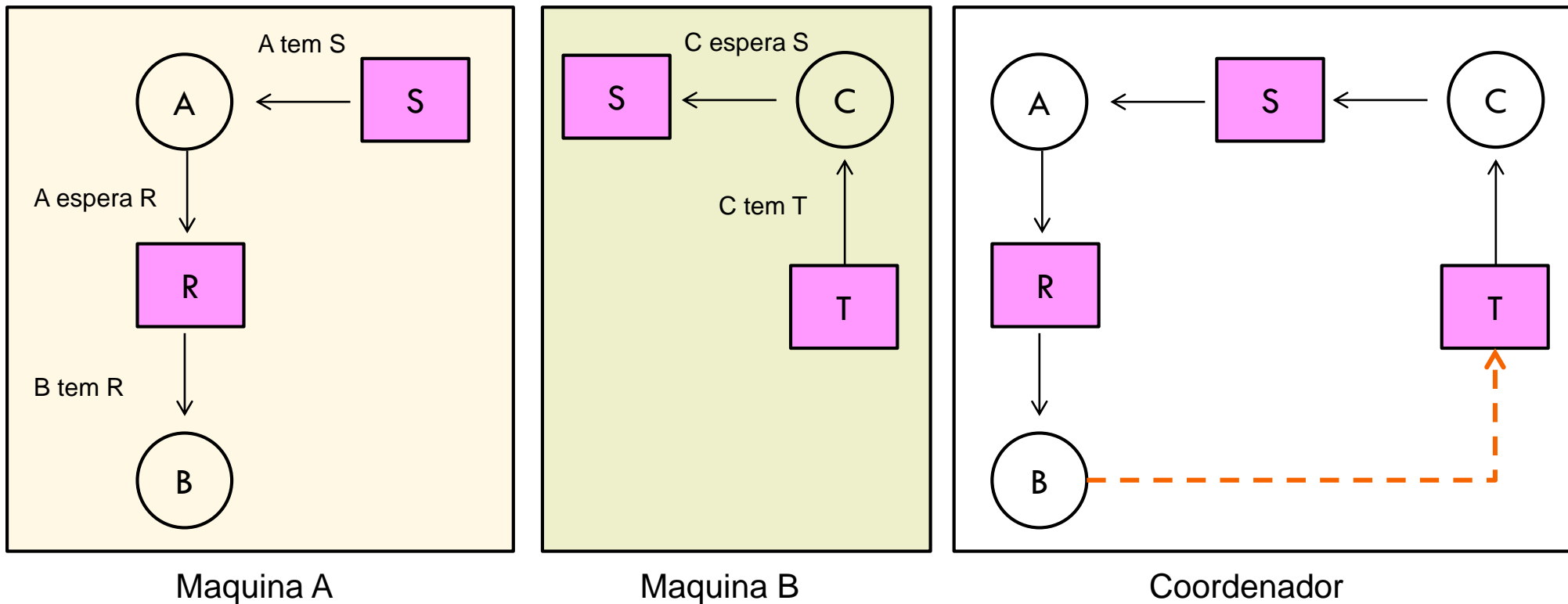


Maquina B



Grafo de espera global

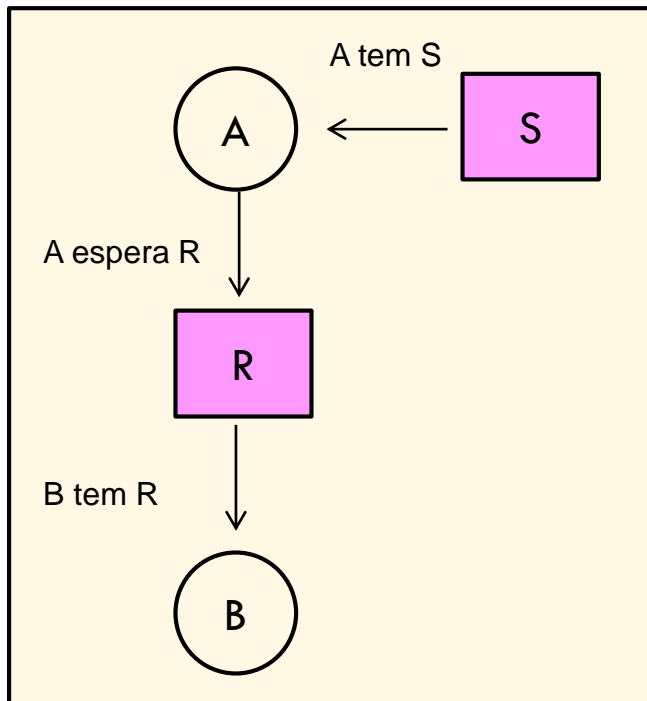
Manutenção da informação (falso deadlock)



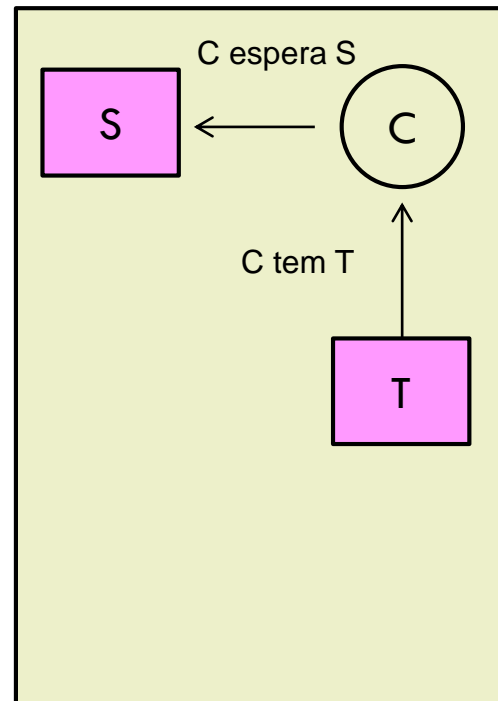
- Máquina A envia mensagem ao coordenador avisando que B liberou o recurso R
- Máquina B envia mensagem ao coordenador avisando que B quer o recurso T

Problema: se a mensagem da maquina B chegar primeiro ocorrerá falso deadlock

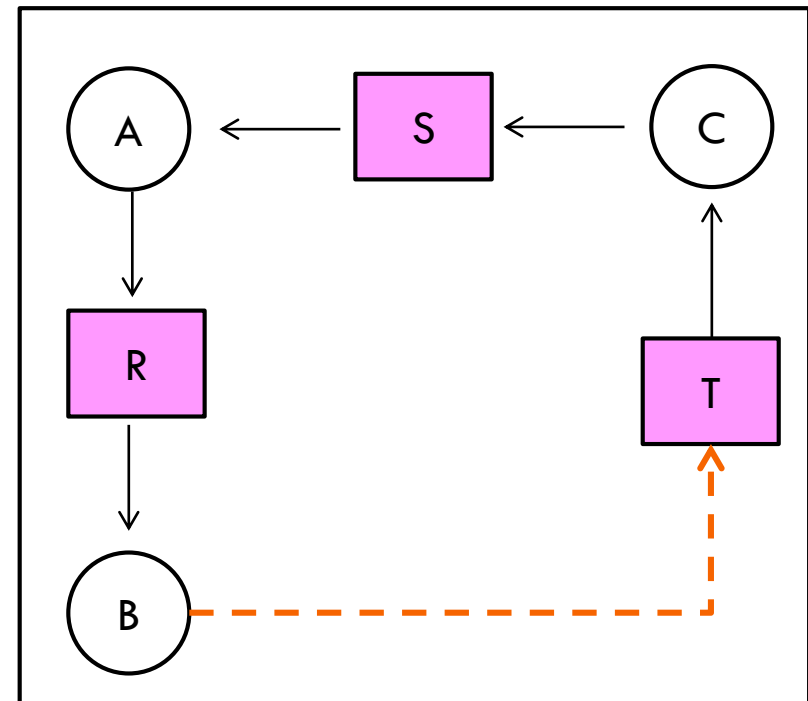
Manutenção da informação (falso deadlock)



Maquina A



Maquina B



Coordenador

- Máquina A envia mensagem ao coordenador avisando que B liberou o recurso R
- Máquina B envia mensagem ao coordenador avisando que B quer o recurso T

Solução: verificar o timestamp

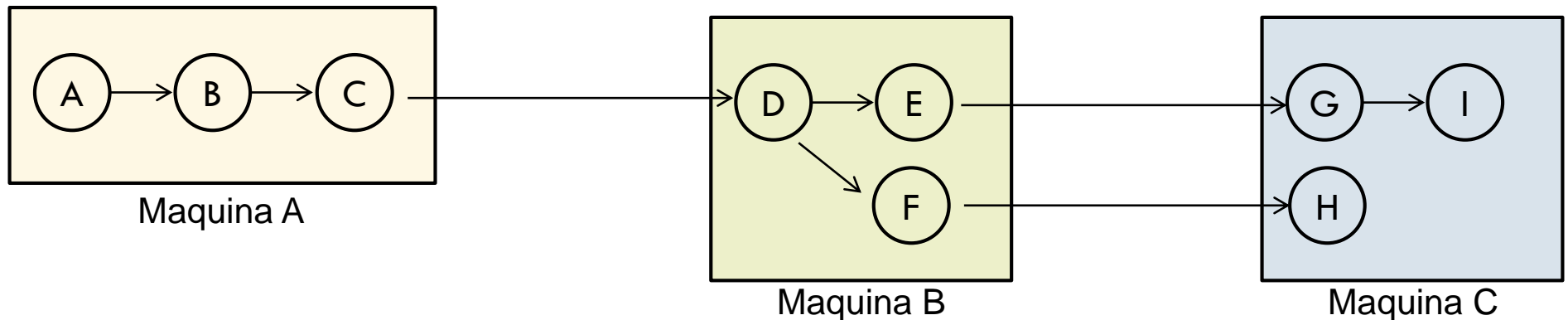
Algoritmo centralizado

Problemas

- Único servidor
 - Escalabilidade
- Custo da transmissão dos grafos locais é alta
- Tempo de atualização do grafo (coordenador pode gerar mais ocorrências de deadlock)

Algoritmo de detecção distribuído

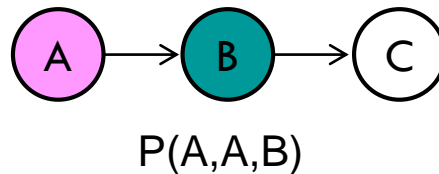
- Algoritmo de Chandy-Misra-Hass (1983)
- Processos estão autorizados a solicitar vários recursos



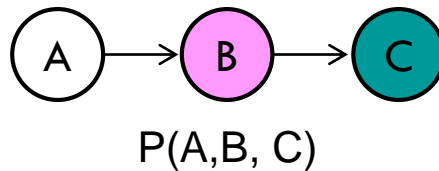
- Uma mensagem é gerada e enviada para processos que possuem os recursos necessários. Esta mensagem possui 3 informações:
 - Número do processo bloqueado
 - Processo que esta enviando a mensagem (receptor)
 - Processo para quem se destina a mensagem (destinatário)

Algoritmo de detecção distribuído

- Exemplo:



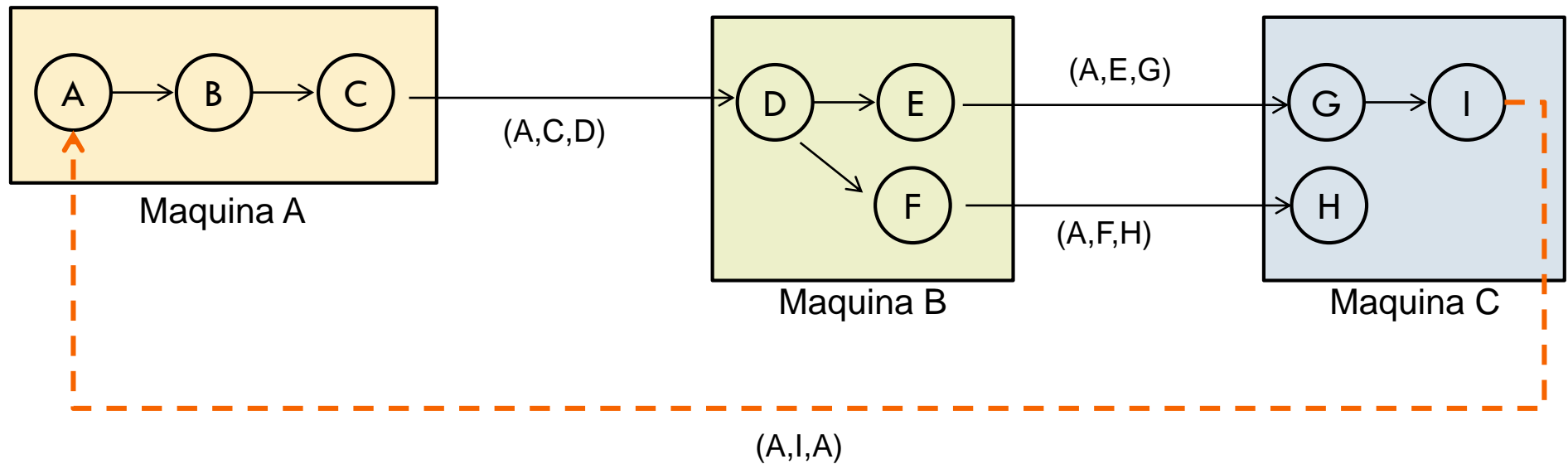
- Ao chegar no receptor, ele verifica se esta esperando algum processo. Se estiver, atualiza o processo que esta enviando e para quem se destina a mensagem



- Se a mensagem voltar para a origem, tem-se deadlock

Algoritmo de detecção distribuído

- Exemplo:



Algoritmo de detecção distribuído

- Formas de se evitar o deadlock
 - Maior identificação
 - Processo que inicia desiste do recurso
 - Problema: se vários processos podem ter iniciado a solicitação de recursos

próxima aula . . .

- Algoritmos de relógios