Desenvolv. de Sistemas Embarcados em Tempo Real

Prof. Hermano Cabral

Departamento de Eletrônica e Sistemas — UFPE

25 de junho de 2024

Plano de Aula

Tema central

- Programação paralela

Plano de Aula

Tema central

- Programação paralela

Objetivos

- Conhecer as características de programação paralela
- Usar mecanismos de sincronização

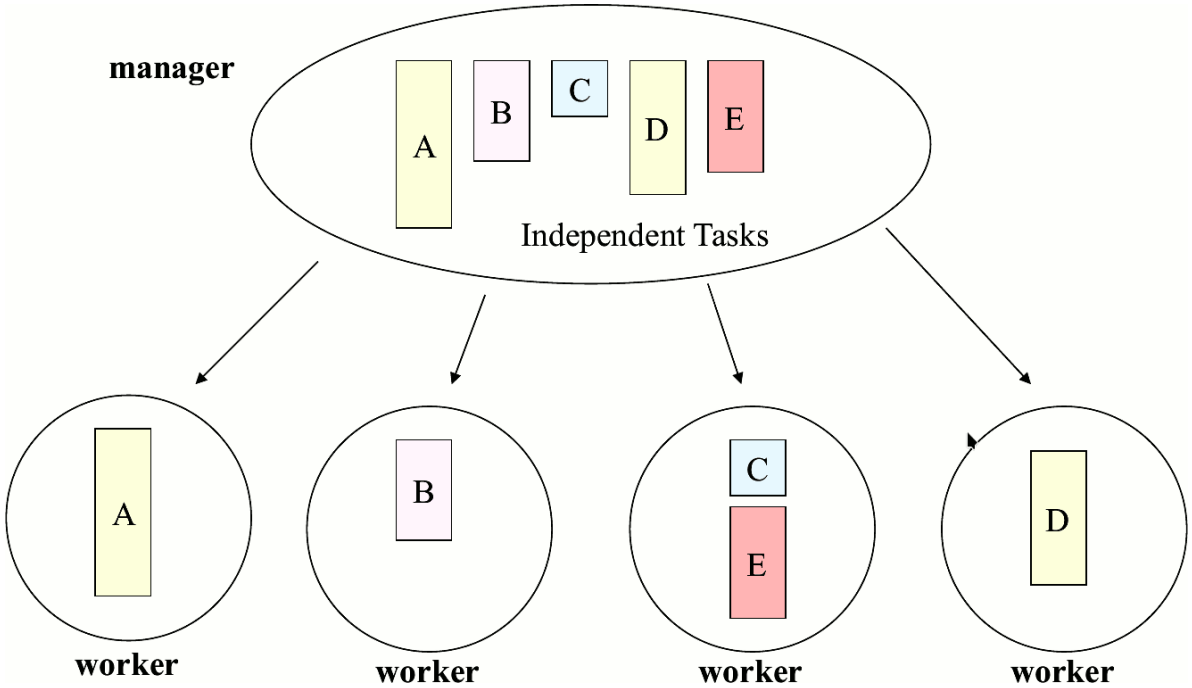
Introdução

- Programação paralela refere-se ao desenvolvimento de programas que executam ao mesmo tempo.

Introdução

- Programação paralela refere-se ao desenvolvimento de programas que executam ao mesmo tempo.
- Ela se preocupa principalmente com 3 aspectos:
  - Eficiência
  - Sincronização
  - Comunicação

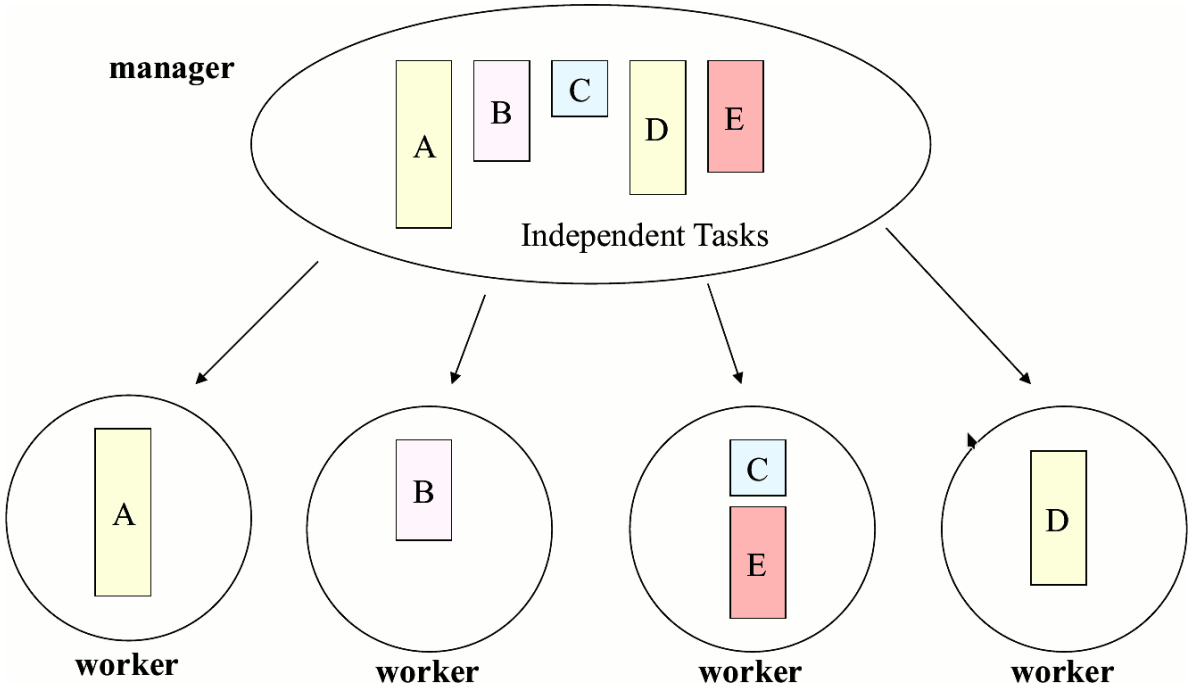
Programação paralela



Eficiência

- Para aumentar a eficiência, procuramos dividir o programa ou os dados em grupos que possam ser executados de forma independente.

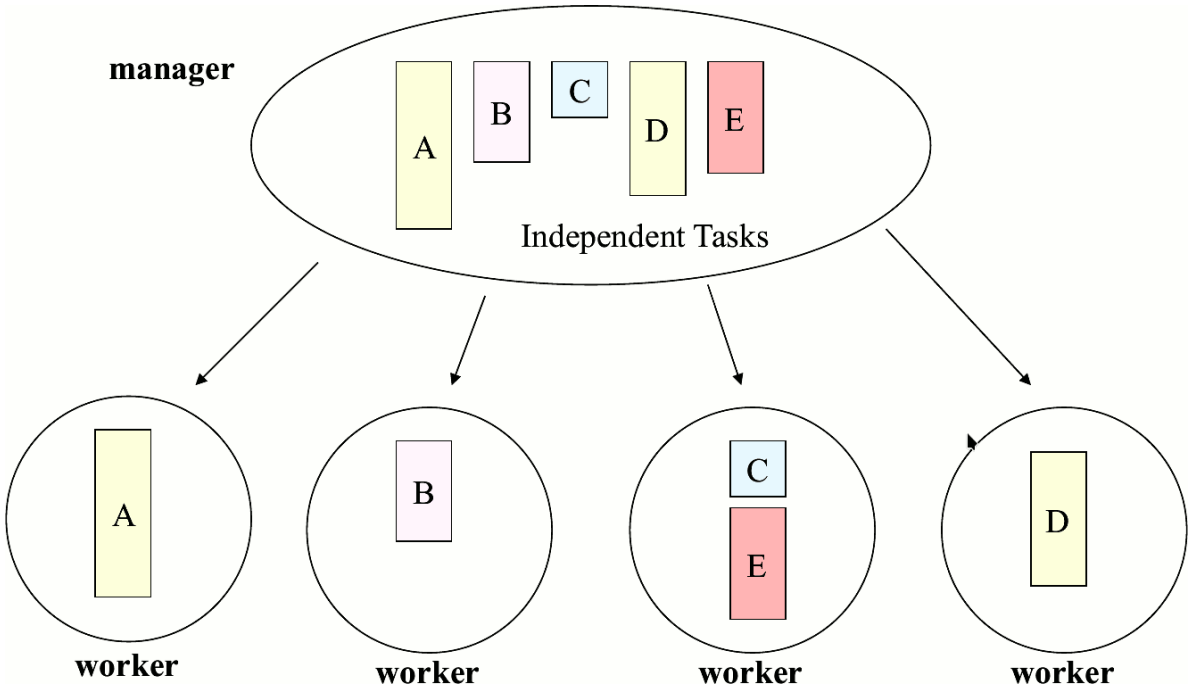
Programação paralela



Eficiência

- Para aumentar a eficiência, procuramos dividir o programa ou os dados em grupos que possam ser executados de forma independente.
- Essa divisão depende do problema que se está resolvendo.

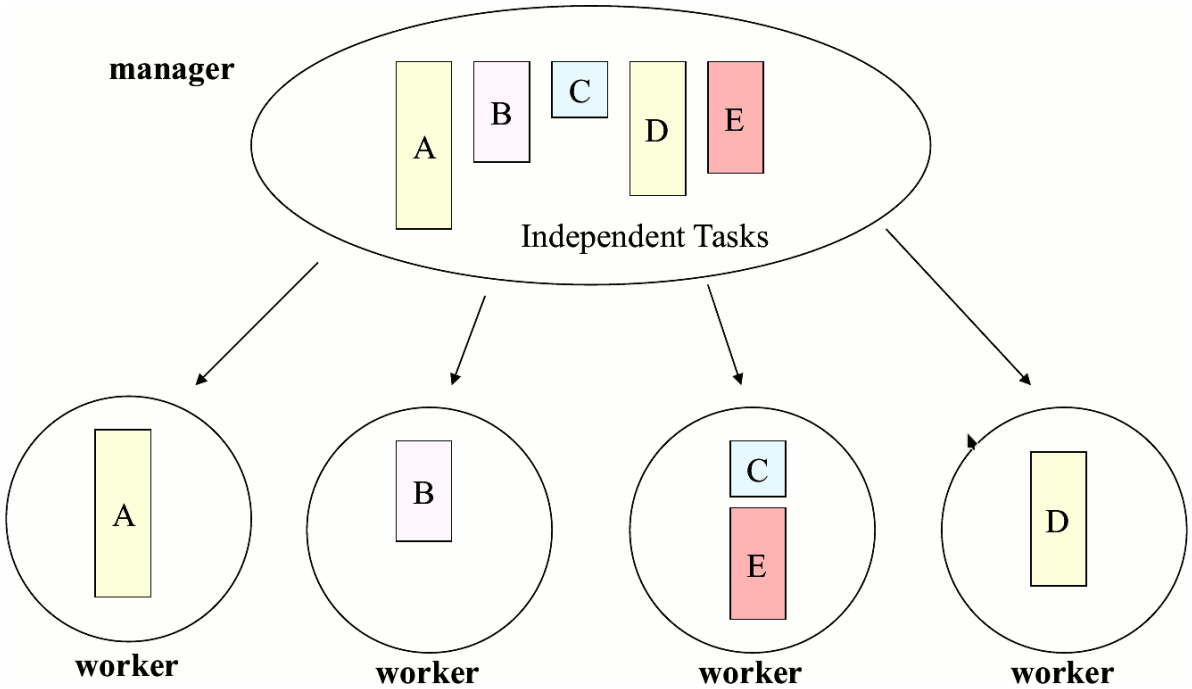
Programação paralela



Eficiência

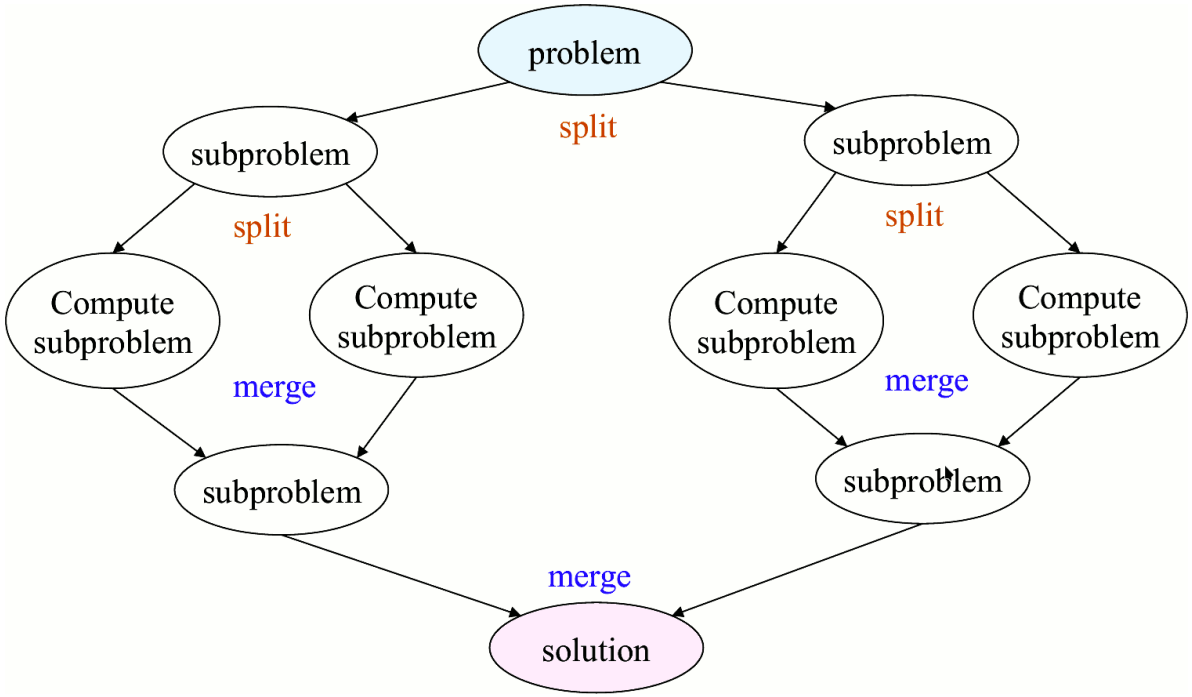
- O melhor caso é aquele que podemos identificar tarefas independentes.

Programação paralela



Eficiência

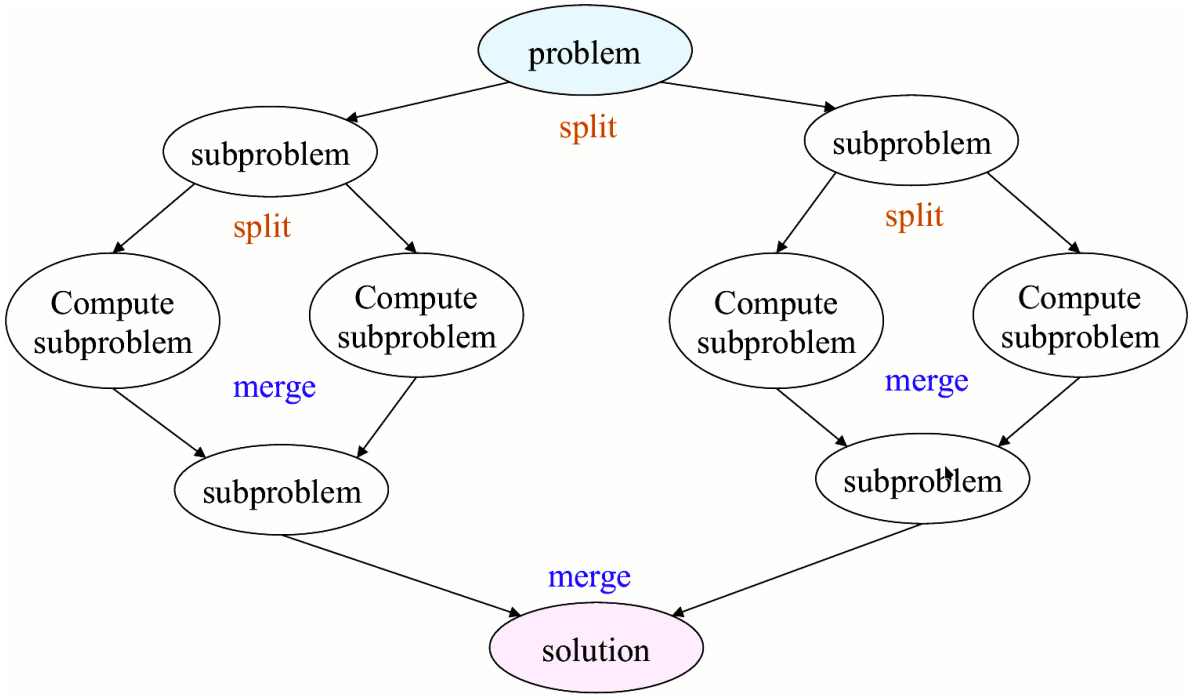
- O melhor caso é aquele que podemos identificar tarefas independentes.
- Isto é mais fácil ao nível *coarse-grain*.



Eficiência

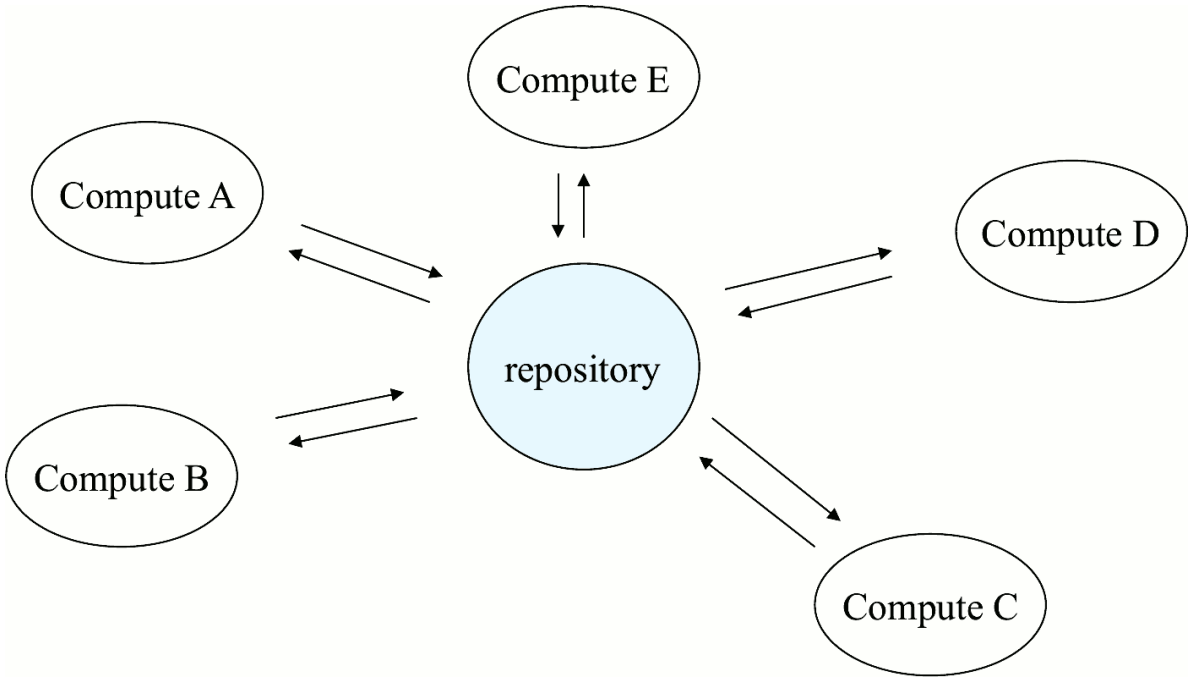
- A um nível mais detalhado (*fine-grained*), o procedimento de divisão-e-conquista pode fornecer mais paralelismo.

Programação paralela



Eficiência

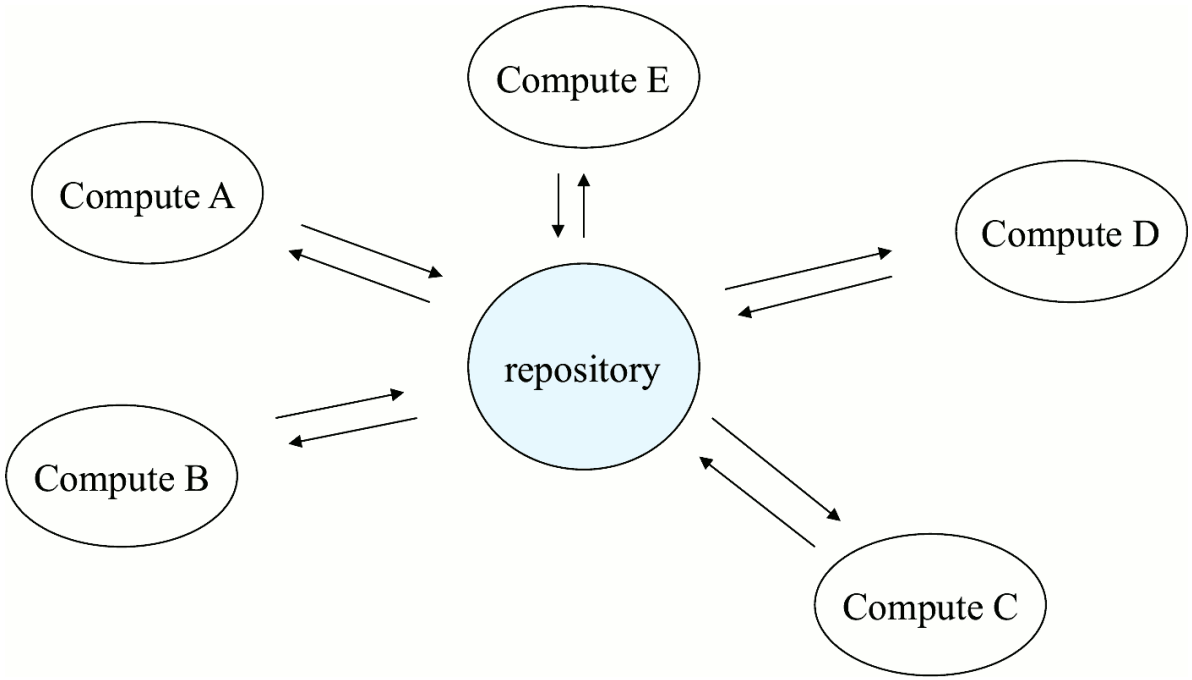
- A um nível mais detalhado (*fine-grained*), o procedimento de divisão-e-conquista pode fornecer mais paralelismo.
- Este procedimento é naturalmente aplicado à máquina de estados.



Sincronismo

- O problema é se as tarefas não forem completamente independentes (o que ocorre com frequência).

Programação paralela



Sincronismo

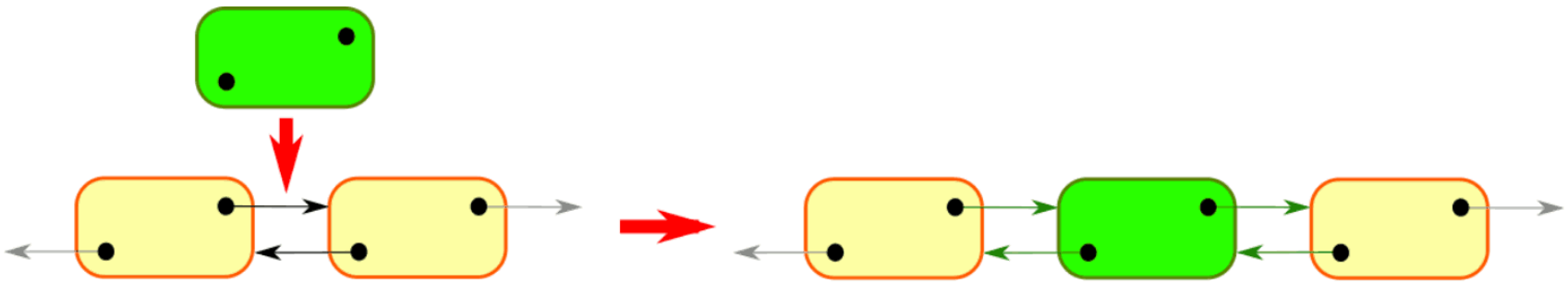
- O problema é se as tarefas não forem completamente independentes (o que ocorre com frequência).
- É necessário mecanismos de sincronização e comunicação.

Sincronismo

- Considere as operações de inserção e remoção de um item em uma lista duplamente encadeada.

Sincronismo

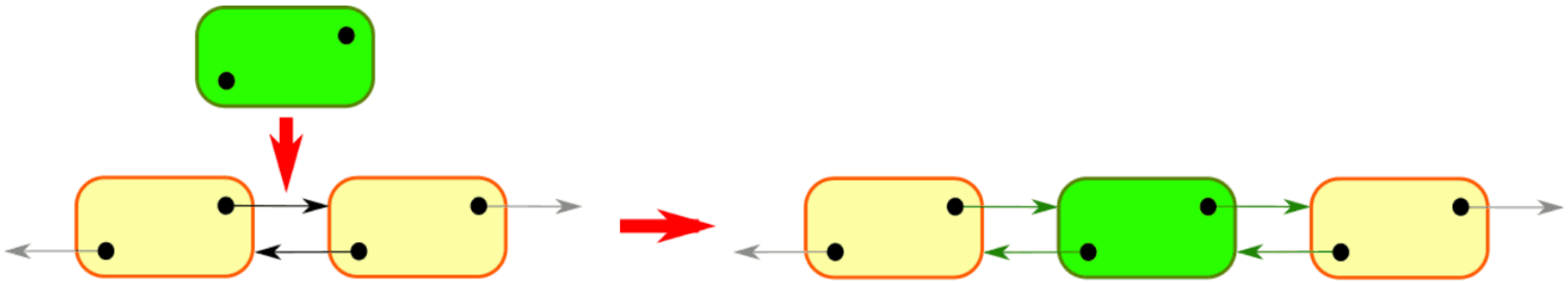
- Considere as operações de inserção e remoção de um item em uma lista duplamente encadeada.
- Cada item da lista tem um ponteiro para o item anterior e outro para o item posterior.



Inserção

Sincronismo

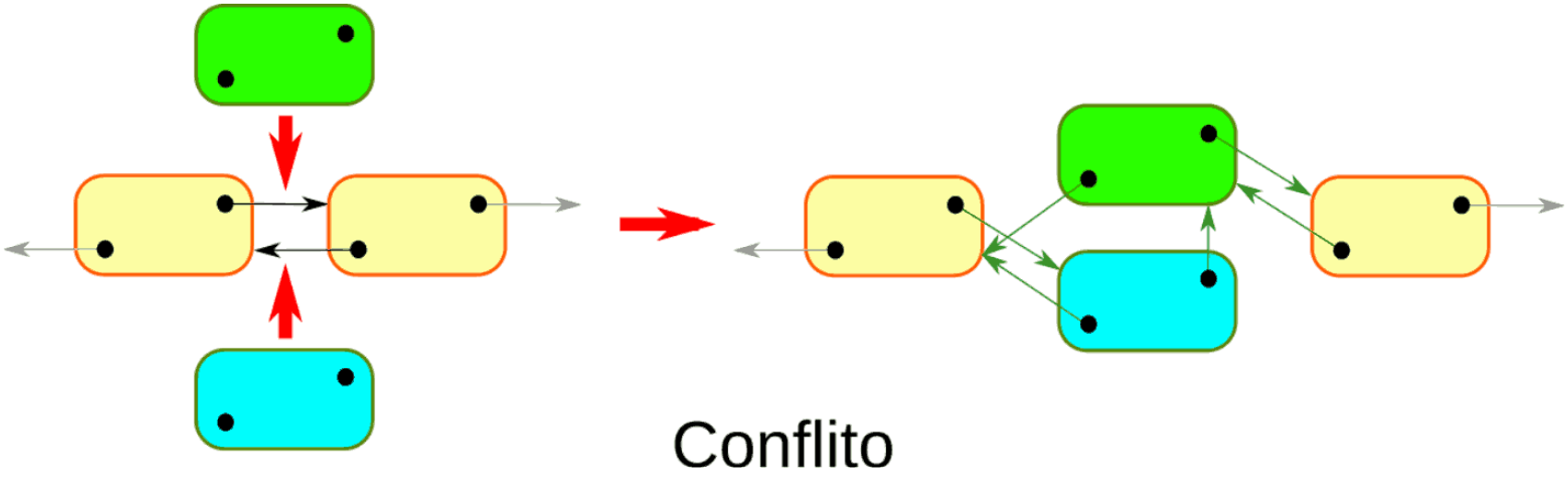
- Para inserção ou remoção, temos as seguintes operações:
  - Atualização do ponteiro da frente do item anterior.
  - Atualização do ponteiro de trás do item posterior.



Inserção

Sincronismo

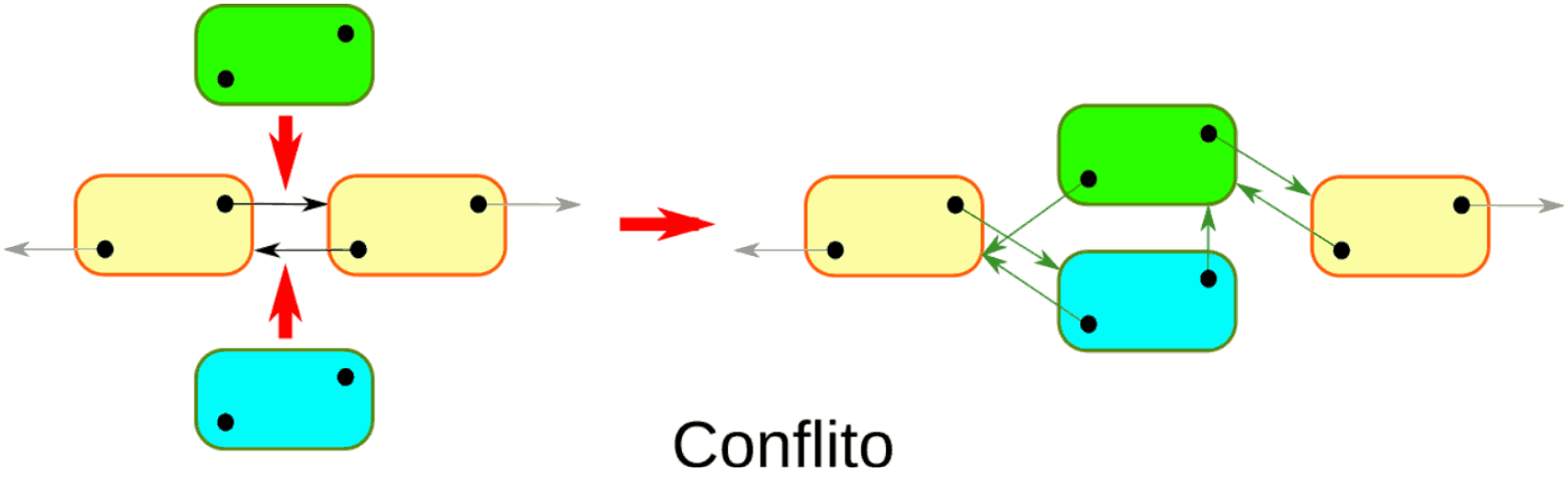
- Para inserção, temos também as seguintes operações:
  - Atualização do ponteiro da frente do novo item
  - Atualização do ponteiro de trás do novo item



Sincronismo

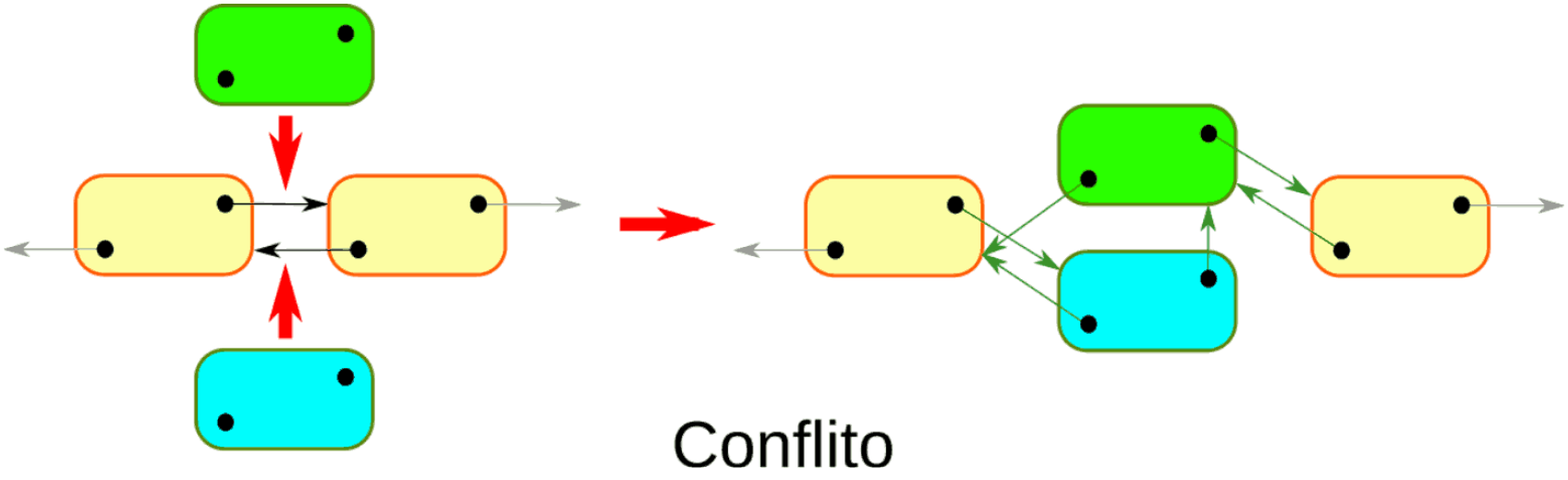
- Suponha que dois itens, V e A, são inseridos ao mesmo tempo.

Sincronismo



Sincronismo

- Suponha que dois itens, V e A, são inseridos ao mesmo tempo.
- Suponha que a ordem das ações é entrelaçada, como V1, A1, A2, A3, A4, V2, V3 e V4.



Sincronismo

- Suponha que dois itens, V e A, são inseridos ao mesmo tempo.
- Suponha que a ordem das ações é entrelaçada, como V1, A1, A2, A3, A4, V2, V3 e V4.
- Um possível resultado está mostrado acima.

Sincronismo

- Um conjunto de operações que devem ser executadas sem interrupção é denominada uma *região crítica*.

Sincronismo

- Um outro exemplo de sincronismo é uma corrida de revezamento onde há a passagem de bastão.

Sincronismo

- Existem alguns mecanismos de sincronismo:
  - Semáforos

Sincronismo

- Existem alguns mecanismos de sincronismo:
  - Semáforos
  - Mutex

Sincronismo — Mutex

- Mutex (Mutual Exclusion) é usado nos casos de proteção de regiões críticas.

Sincronismo — Mutex

- Mutex (Mutual Exclusion) é usado nos casos de proteção de regiões críticas.
- As operações em um mutex são o seu travamento e destravamento.

Sincronismo — Mutex

- Mutex (Mutual Exclusion) é usado nos casos de proteção de regiões críticas.
- As operações em um mutex são o seu travamento e destravamento.
- Uma mesma tarefa trava e posteriormente destrava.

Sincronismo — Semáforos

- Semáforos são usados em casos de sincronismo entre duas tarefas.

Sincronismo — Semáforos

- Semáforos são usados em casos de sincronismo entre duas tarefas.
- As operações em um semáforo são a espera e a sinalização.

Sincronismo — Semáforos

- Semáforos são usados em casos de sincronismo entre duas tarefas.
- As operações em um semáforo são a espera e a sinalização.
- Uma tarefa espera a sinalização que outra fará.

Comunicação

- Muitas vezes, as tarefas precisam não apenas sincronizar, mas também se comunicar.

Comunicação

- Muitas vezes, as tarefas precisam não apenas sincronizar, mas também se comunicar.
- Existem alguns mecanismos de comunicação:
  - Eventos
  - Mailboxes
  - Memória compartilhada

Problemas

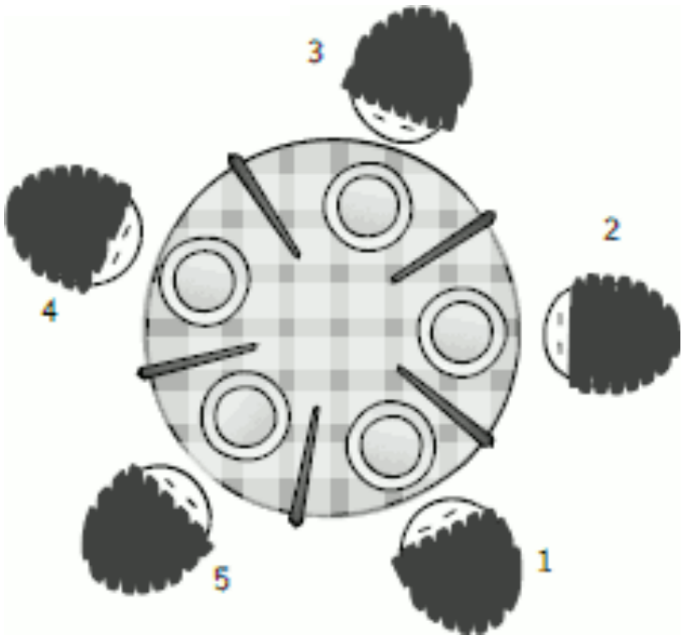
- Existem algumas situações de sincronismo que devem ser evitadas:
  - Impasse (*deadlock*)

Problemas

- Existem algumas situações de sincronismo que devem ser evitadas:
  - Impasse (*deadlock*)
  - Tarefas famintas (starvation)

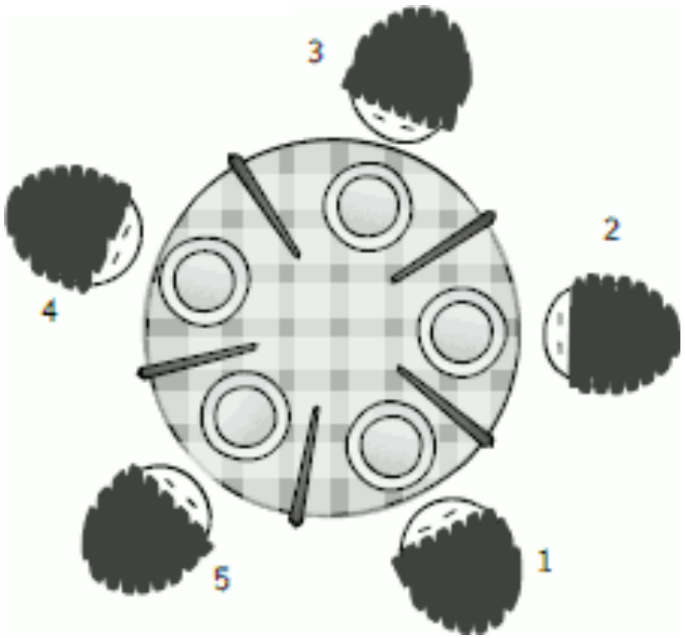
Problemas

- Existem algumas situações de sincronismo que devem ser evitadas:
  - Impasse (*deadlock*)
  - Tarefas famintas (starvation)
  - Condição de corrida (Race conditions)



Problemas — impasse

- Esta situação ocorre quando 2 tarefas esperam por um recurso que a outra está retendo.



Problemas — impasse

- Esta situação ocorre quando 2 tarefas esperam por um recurso que a outra está retendo.
- Pode ser resolvido alocando os recursos em uma ordem específica.

Problemas — tarefas famintas

- Esta situação ocorre quando uma tarefa se apodera por tempo demasiado de um recurso, impedindo outras tarefas de acessar o recurso.

Problemas — tarefas famintas

- Esta situação ocorre quando uma tarefa se apodera por tempo demasiado de um recurso, impedindo outras tarefas de acessar o recurso.
- A solução para isto é uma partilha justa do recurso entre as tarefas.

Problemas — condição de corrida

- Este problema ocorre quando uma sequência de ações é executada de forma paralela por 2 ou mais tarefas e não há sincronismo entre elas.

Problemas — condição de corrida

- Este problema ocorre quando uma sequência de ações é executada de forma paralela por 2 ou mais tarefas e não há sincronismo entre elas.
- Isto é resolvido com uma sincronização correta entre as tarefas.

Dicas para a disciplina

- Use Mutex para proteger recursos compartilhados.

Dicas para a disciplina

- Use Mutex para proteger recursos compartilhados.
- Use semáforos ou eventos para se comunicar ou sincronizar entre threads.

Dicas para a disciplina

- Use Mutex para proteger recursos compartilhados.
- Use semáforos ou eventos para se comunicar ou sincronizar entre threads.
- Seja o mais rápido e breve possível nas threads de maior prioridade.

Dicas para a disciplina

- Use Mutex para proteger recursos compartilhados.
- Use semáforos ou eventos para se comunicar ou sincronizar entre threads.
- Seja o mais rápido e breve possível nas threads de maior prioridade.
- Caso necessite obter 2 ou mais recursos protegidos em 2 ou mais lugares de seu código, sempre use a mesma ordem de obtenção.