Comunicação entre processos e threads

Contexto

- Atender vários usuários simultaneamente
- Computadores Multicore (tarefas que cooperam)
- Modularidade
- Aplicações interativas

Comunicação

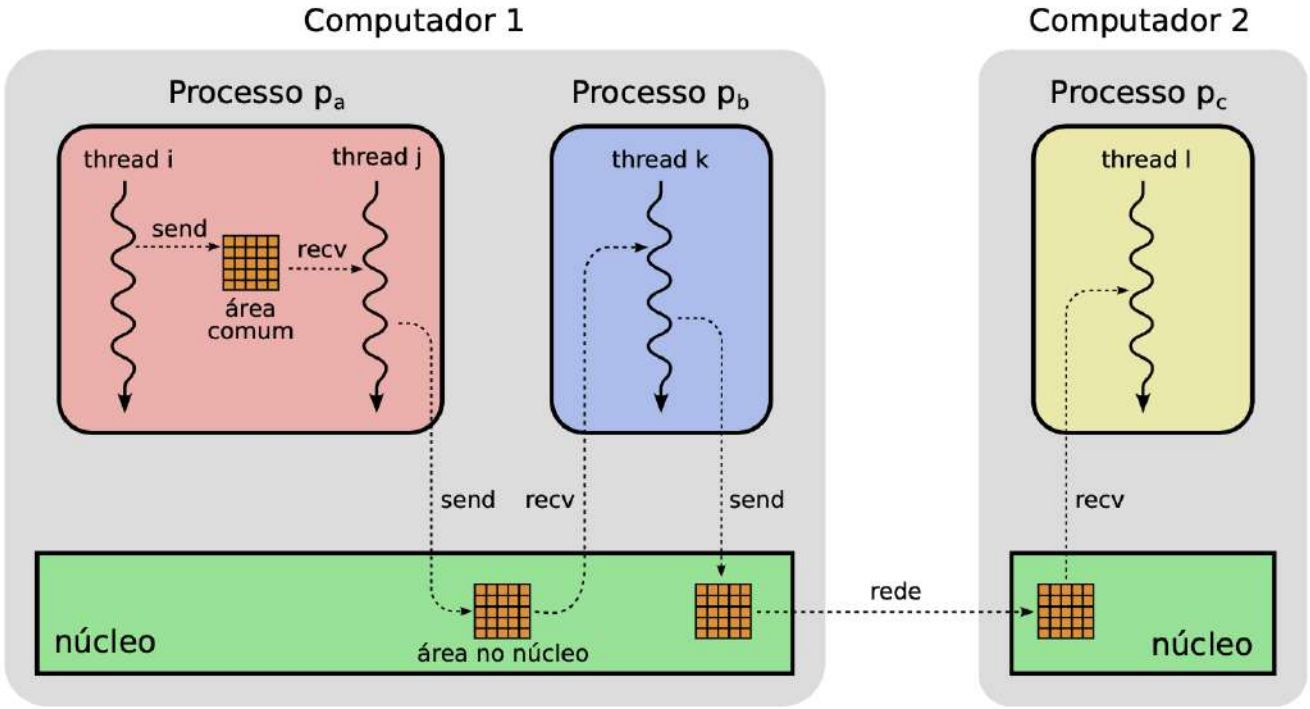


Figura 8.1: Comunicação intraprocesso ($t_i \rightarrow t_j$), interprocessos ($t_j \rightarrow t_k$) e intersistemas ($t_k \rightarrow t_l$).

Aspectos de Comunicação

- **Direta**(o destino é o processo) x **Indireta** (canal é o destino)
- Sincronismo
  - Bloqueante
  - Não bloqueante (assíncrona)
- Formato de Envio
  - Sequência de Msg independentes
  - Fluxo sequencial
- Capacidade dos Canais (*buffers)*
  - *Nula*
  - Infinita
  - Finita

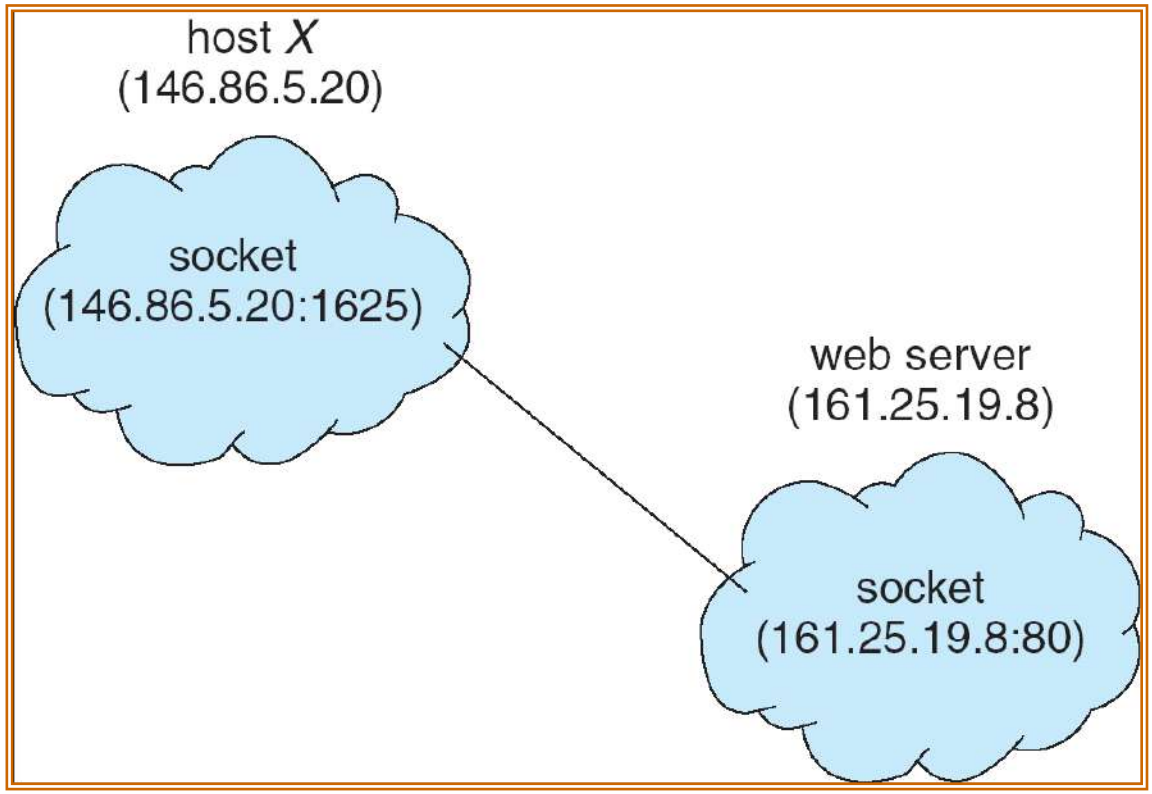
Aspectos de Comunicação

- Confiabilidade de canais (informação e sequência)
  - Perda de dados
  - Perda da integridade
  - Perda da ordem
- Número de participantes
  - 1:1
  - M:N
  - *Nula*
  - Infinita
  - Finita

Soquetes

- Um soquete é definido como uma *extremidade para comunicação*
- Concatenação de endereço IP e porta
- O soquete **161.25.19.8:1625** refere-se à porta **1625** no host **161.25.19.8**
- A comunicação consiste entre um par de soquetes

Comunicação por soquete

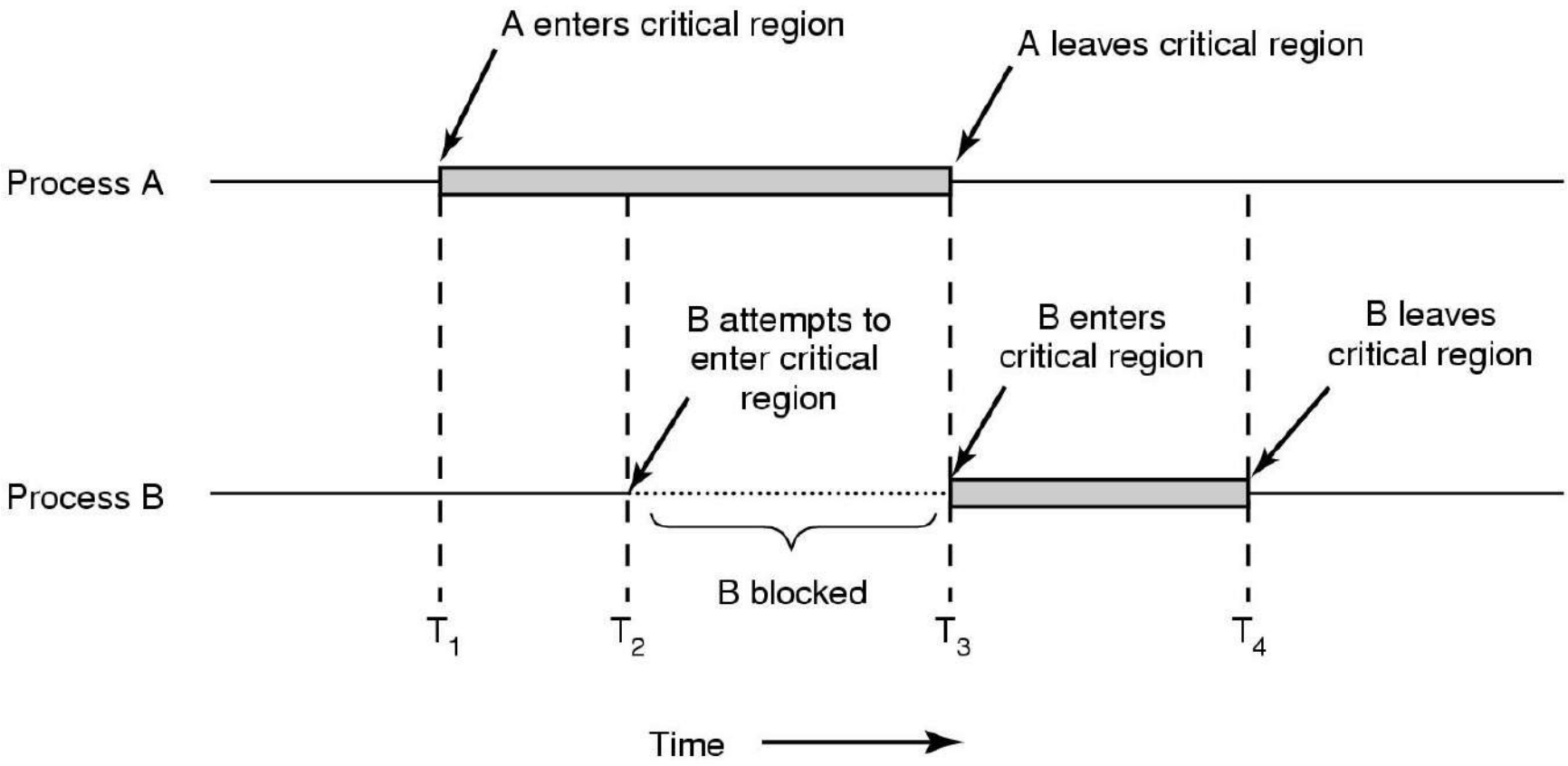


Sincronismo entre processos

Problema de seção crítica

1. **Condição race/condição de corrida** – Quando há acesso simultâneo aos dados compartilhados e o resultado final depende da ordem de execução.
2. **Seção crítica** – Seção do código onde os dados compartilhados são acessados.
3. **Seção de entrada** - Código que solicita permissão para entrar em sua seção crítica.
4. **Seção de saída** – Código executado após a saída da seção crítica.

Região Crítica



Requisitos para solução (seção crítica)

- **Exclusão mútua**
  - Se um processo Pi está executando sua região crítica nenhum outro poderá executar a sua região crítica
- **Progresso**
  - Caso não exista nenhum processo executando a sua seção crítica e alguns processos queriam entrar em suas seções críticas, apenas processos que não estejam executando suas seções remanescentes podem participar da "decisão";
  - Assim, nenhum processo fora de sua região crítica pode bloquear outro processo
  - Esta decisão não pode ser adiada infinitamente;

Requisitos para solução (seção crítica)

- **Espera limitada**
  - Limite de quantas vezes outros processos podem entrar em suas seções críticas após um processo ter feito solicitação para entrar em sua seção crítica e antes dessa solicitação ser atendida

Semáforo

- Ferramenta de sincronismo que não exige espera ocupada
- Semáforo *S* – variável inteira
- Duas operações padrão modificam S: acquire() e release()
  - Originalmente chamadas P() e V()
- Menos complicado
- Só pode ser acessado por duas operações indivisíveis

Semáforo como ferramenta geral de sincronismo

- Semáforo de contagem – valor inteiro pode variar por um domínio irrestrito
- Semáforo binário – valor inteiro só pode variar entre o e 1; pode ser mais simples de implementar
  - Também conhecidos como locks mutex

Deadlock e starvation

- **Deadlock** – dois ou mais processos estão esperando indefinidamente por um evento que só pode ser causado somente por um dos processos que está esperando

- **Starvation/inanição** – *lock* indefinido. Um processo pode nunca ser removido da fila de semáforo em que ele é suspenso.