



UNIVERSIDADE  
FEDERAL DE  
SERGIPE



DEPARTAMENTO  
DE COMPUTAÇÃO

# Multiprocessamento

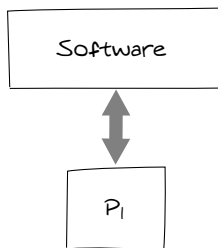
## Arquitetura de Computadores

Bruno Prado

Departamento de Computação / UFS

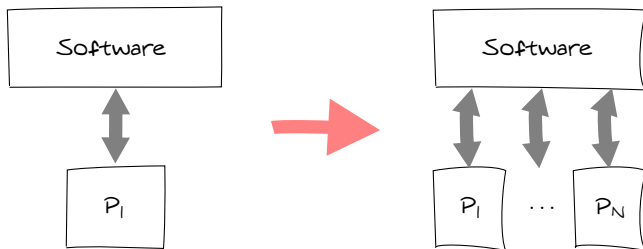
# Introdução

- ▶ O que é multiprocessamento?
  - ▶ É uma organização que utiliza múltiplas unidades de processamento para a execução do software
  - ▶ Este paradigma emergiu da exaustão da capacidade de paralelismo em nível de instrução



# Introdução

- ▶ O que é multiprocessamento?
  - ▶ É uma organização que utiliza múltiplas unidades de processamento para a execução do software
  - ▶ Este paradigma emergiu da exaustão da capacidade de paralelismo em nível de instrução



# Introdução

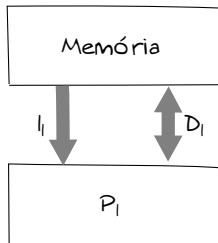
- ▶ Categorização de fluxos de instruções e de dados
  - ▶ *Single Instruction Single Data* (SISD)
  - ▶ *Single Instruction Multiple Data* (SIMD)
  - ▶ *Multiple Instruction Single Data* (MISD)
  - ▶ *Multiple Instruction Multiple Data* (MIMD)

	Única instrução	Múltiplas instruções
Único dado	SISD	MISD
Múltiplos dados	SIMD	MIMD

# Introdução

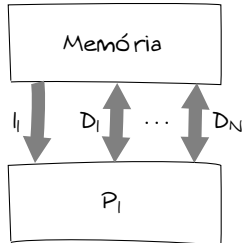
- ▶ *Single Instruction Single Data (SISD)*

- ▶ Uma única instrução é buscada na memória e executada no processador por vez
- ▶ Somente um dado da memória pode ser acessado para escrita ou leitura em cada operação realizada



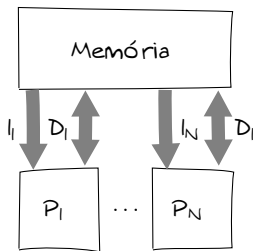
# Introdução

- ▶ *Single Instruction Multiple Data (SIMD)*
  - ▶ A instrução em execução no processador é capaz manipular múltiplos dados simultaneamente
  - ▶ Este tipo de organização permite a execução sequencial da instrução com acesso e processamento paralelo dos dados
  - ▶ Extensões multimídia dos processadores



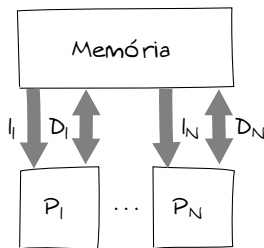
# Introdução

- ▶ *Multiple Instruction Single Data (MISD)*
  - ▶ Uma sequência de instruções são executadas por múltiplos processadores executando sobre um mesmo conjunto de dados (redundância)
  - ▶ Apesar de ser uma organização possível, é utilizada apenas em sistemas especializados tolerantes a falhas



# Introdução

- ▶ *Multiple Instruction Multiple Data (MIMD)*
  - ▶ Os múltiplos processadores executam diferentes sequências de instruções operando sobre conjuntos distintos de dados centralizados ou distribuídos
  - ▶ É o multiprocessamento de propósito geral



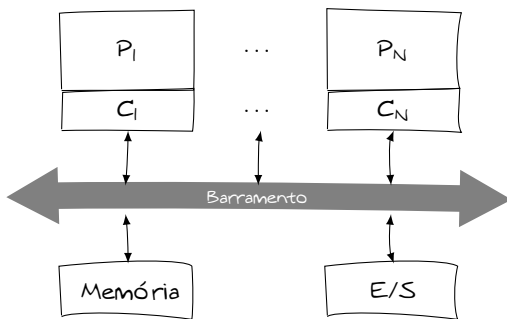


# Introdução

- ▶ Como é feita a comunicação entre os processadores em uma organização multiprocessada?
  - ▶ Memória compartilhada
    - ▶ Cada processador acessa as instruções e os dados armazenados em uma mesma memória com tempo de acesso uniforme (UMA) e a troca de informações entre eles é feita por variáveis compartilhadas

# Introdução

- ▶ Como é feita a comunicação entre os processadores em uma organização multiprocessada?
  - ▶ Memória compartilhada
    - ▶ Cada processador acessa as instruções e os dados armazenados em uma mesma memória com tempo de acesso uniforme (UMA) e a troca de informações entre eles é feita por variáveis compartilhadas



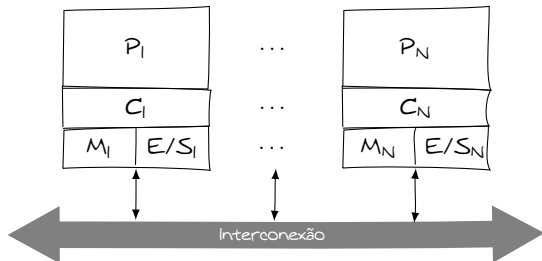
Multiprocessamento simétrico (SMP)

# Introdução

- ▶ Como é feita a comunicação entre os processadores em uma organização multiprocessada?
  - ▶ Memória distribuída
    - ▶ Os processadores possuem seu próprio espaço de endereçamento para acesso não uniforme a memória (NUMA) e realizando a comunicação entre si de forma explícita através da troca de mensagens

# Introdução

- ▶ Como é feita a comunicação entre os processadores em uma organização multiprocessada?
  - ▶ Memória distribuída
    - ▶ Os processadores possuem seu próprio espaço de endereçamento para acesso não uniforme a memória (NUMA) e realizando a comunicação entre si de forma explícita através da troca de mensagens



Multiprocessamento assimétrico (AMP)

# Multiprocessamento

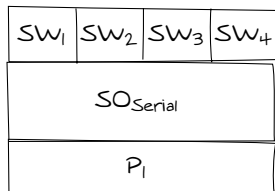
- ▶ Multiprocessamento simétrico (SMP)
  - ▶ Problema
    - ▶ Limitações cada vez maiores do paradigma ILP
    - ▶ Crescente demanda por desempenho dos sistemas

# Multiprocessamento

- ▶ Multiprocessamento simétrico (SMP)
  - ▶ Problema
    - ▶ Limitações cada vez maiores do paradigma ILP
    - ▶ Crescente demanda por desempenho dos sistemas
  - ▶ Contexto
    - ▶ Existe uma grande quantidade de software desenvolvido para executar em plataformas uniprocessadas
    - ▶ É muito complexa a modificação de todo o software existente para suportar múltiplos processadores

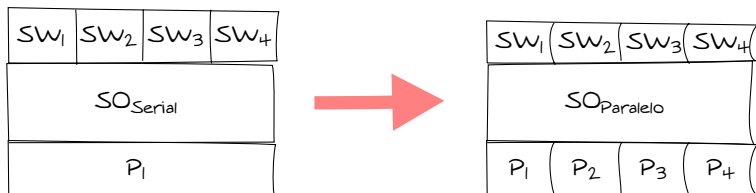
# Multiprocessamento

- ▶ Multiprocessamento simétrico (SMP)
  - ▶ O conceito de multiprocessamento simétrico é uma resposta para o problema de demanda por desempenho e compatibilidade com o software desenvolvido para execução uniprocessada



# Multiprocessamento

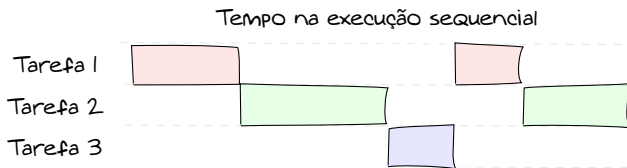
- ▶ Multiprocessamento simétrico (SMP)
  - ▶ O conceito de multiprocessamento simétrico é uma resposta para o problema de demanda por desempenho e compatibilidade com o software desenvolvido para execução uniprocessada





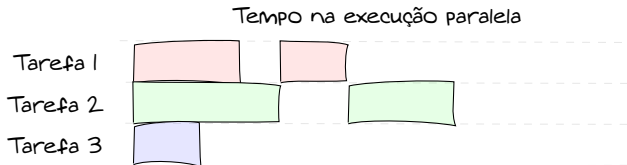
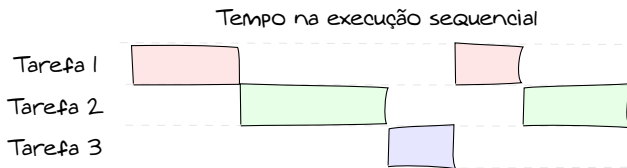
# Multiprocessamento

- ▶ Multiprocessamento simétrico (SMP)
  - ▶ Cada processo do sistema possui seu próprio espaço de endereçamento na memória virtual e o sistema operacional faz o escalonamento dos processos entre os núcleos de processamento disponíveis



# Multiprocessamento

- ▶ Multiprocessamento simétrico (SMP)
  - ▶ Cada processo do sistema possui seu próprio espaço de endereçamento na memória virtual e o sistema operacional faz o escalonamento dos processos entre os núcleos de processamento disponíveis



# Multiprocessamento

- ▶ Multiprocessamento simétrico (SMP)
  - ▶ Características chave
    - ▶ Existem pelo menos dois processadores com instruções compatíveis e com recursos de hardware similares

# Multiprocessamento

- ▶ Multiprocessamento simétrico (SMP)
  - ▶ Características chave
    - ▶ Existem pelo menos dois processadores com instruções compatíveis e com recursos de hardware similares
    - ▶ Estes processadores compartilham o mesmo espaço de endereçamento físico da memória principal e para os recursos de E/S, utilizando um barramento

# Multiprocessamento

- ▶ Multiprocessamento simétrico (SMP)
  - ▶ Características chave
    - ▶ Existem pelo menos dois processadores com instruções compatíveis e com recursos de hardware similares
    - ▶ Estes processadores compartilham o mesmo espaço de endereçamento físico da memória principal e para os recursos de E/S, utilizando um barramento
    - ▶ Todos os processadores possuem a mesma função ou papel na plataforma (simétricos)

# Multiprocessamento

- ▶ Multiprocessamento simétrico (SMP)
  - ▶ Vantagens potenciais
    - ▶ Aumento do desempenho com execução paralela do software através dos processos do sistema

# Multiprocessamento

- ▶ Multiprocessamento simétrico (SMP)
  - ▶ Vantagens potenciais
    - ▶ Aumento do desempenho com execução paralela do software através dos processos do sistema
    - ▶ Disponibilidade do sistema é melhorada em caso de falha de um dos núcleos de processamento

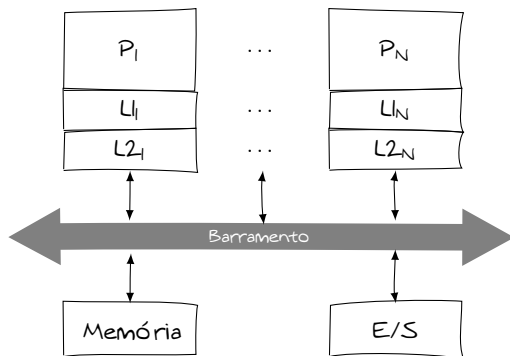
# Multiprocessamento

- ▶ Multiprocessamento simétrico (SMP)
  - ▶ Vantagens potenciais
    - ▶ Aumento do desempenho com execução paralela do software através dos processos do sistema
    - ▶ Disponibilidade do sistema é melhorada em caso de falha de um dos núcleos de processamento
    - ▶ Escalabilidade com crescimento incremental, permitindo diferentes quantidades de processadores



# Multiprocessamento

- ▶ Multiprocessamento simétrico (SMP)
  - ▶ Como cada processador é um mestre do barramento, são necessárias técnicas de arbitração para tratar as requisições concorrentes, o que pode causar retenção no tráfego de informações
  - ▶ Para reduzir o tráfego é utilizada uma hierarquia de cache com múltiplos níveis com escrita atrasada



# Multiprocessamento

- ▶ Multiprocessamento simétrico (SMP)
  - ▶ Hierarquia de cache
    - ▶ Cada nível da hierarquia representa uma cópia do conteúdo armazenado na memória principal

# Multiprocessamento

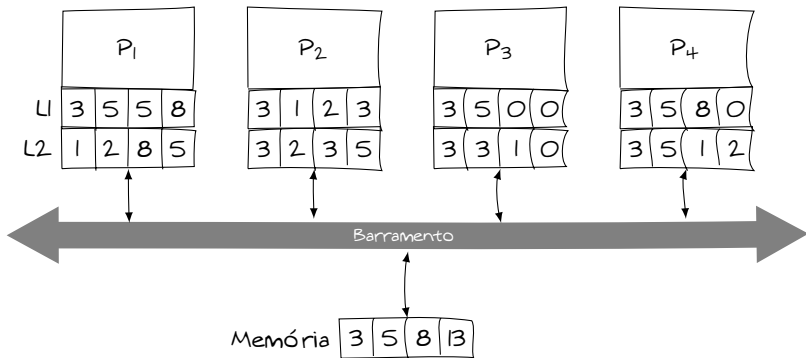
- ▶ Multiprocessamento simétrico (SMP)
  - ▶ Hierarquia de cache
    - ▶ Cada nível da hierarquia representa uma cópia do conteúdo armazenado na memória principal
    - ▶ Para a execução concorrente do software pelos processadores, é necessário que variáveis compartilhadas em memória sejam utilizadas

# Multiprocessamento

- ▶ Multiprocessamento simétrico (SMP)
  - ▶ Hierarquia de cache
    - ▶ Cada nível da hierarquia representa uma cópia do conteúdo armazenado na memória principal
    - ▶ Para a execução concorrente do software pelos processadores, é necessário que variáveis compartilhadas em memória sejam utilizadas
    - ▶ Como cada processador possui sua própria cópia da memória em seus níveis de cache, é possível que os mesmos endereços de memória estejam mapeados em blocos com diferentes valores (inconsistência)

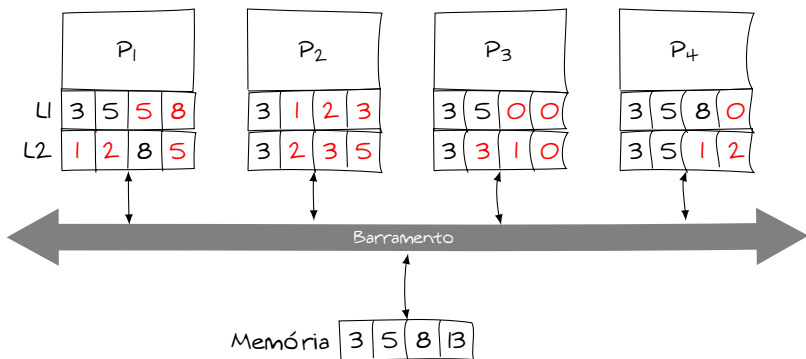
# Multiprocessamento

- ▶ Multiprocessamento simétrico (SMP)
  - ▶ Hierarquia de cache



# Multiprocessamento

- ▶ Multiprocessamento simétrico (SMP)
  - ▶ Hierarquia de cache



Inconsistência de dados

# Multiprocessamento

- ▶ Coerência de cache
  - ▶ São técnicas aplicadas em tempo de execução para tratar condições de inconsistência dos dados
  - ▶ Estes mecanismos são baseados em protocolos para transmissão de metadados das caches
    - ▶ Distribuído com protocolo *snoopy*
    - ▶ Centralizado com utilização de diretório

# Multiprocessamento

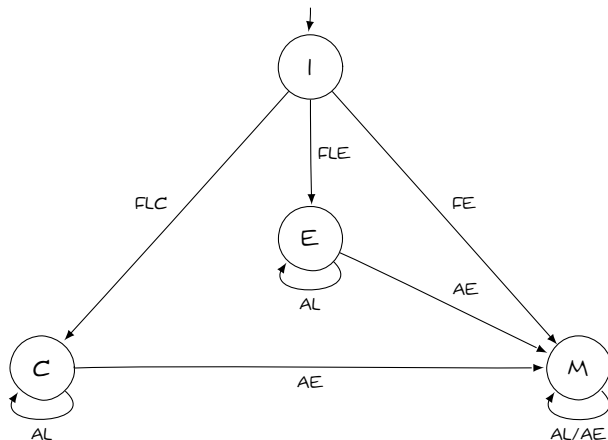
- ▶ Coerência de cache
  - ▶ Campos de metadados

Campo	Descrição
C	Outras caches podem possuir cópias deste mesmo conjunto de dados, mas os dados são idênticos aos armazenados na memória
E	Os dados contidos nesta cache não estão presentes em nenhuma outra cache e não possuem modificações com relação a memória
I	Este campo é utilizado para marcar se uma determinada linha da cache foi invalidada ou possui dados que não válidos
M	Quando é feita uma modificação na cache com política de escrita atrasada, este campo sinaliza que o dado difere do armazenado na memória



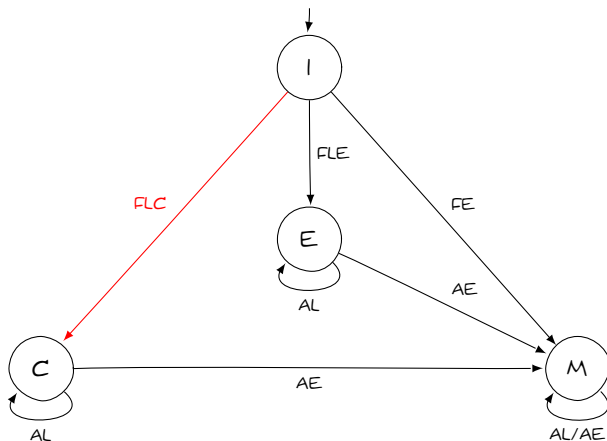
# Multiprocessamento

- Coerência de cache
  - Campos de metadados



# Multiprocessamento

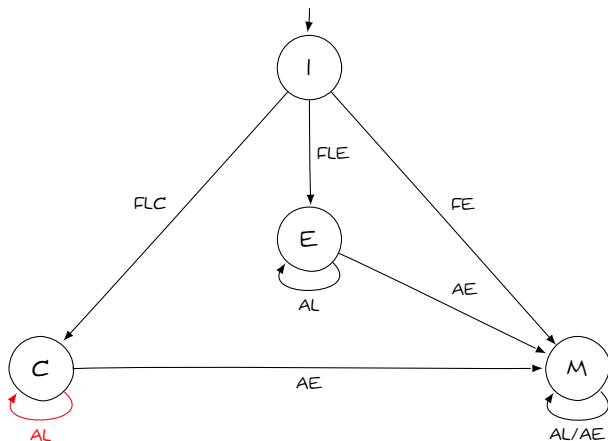
- Coerência de cache
  - Campos de metadados



Falta de leitura compartilhada (FLC)

# Multiprocessamento

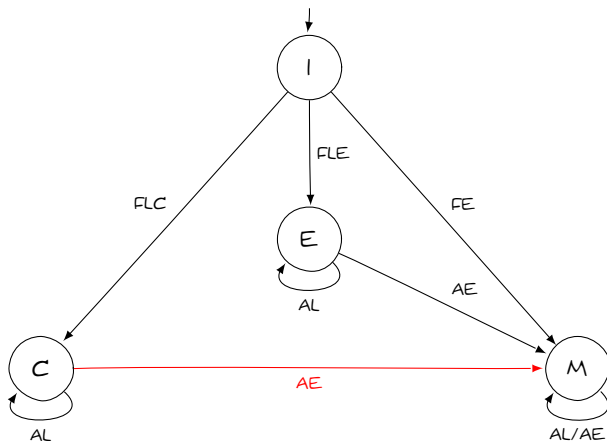
- Coerência de cache
  - Campos de metadados



Acerto de leitura (AL)

# Multiprocessamento

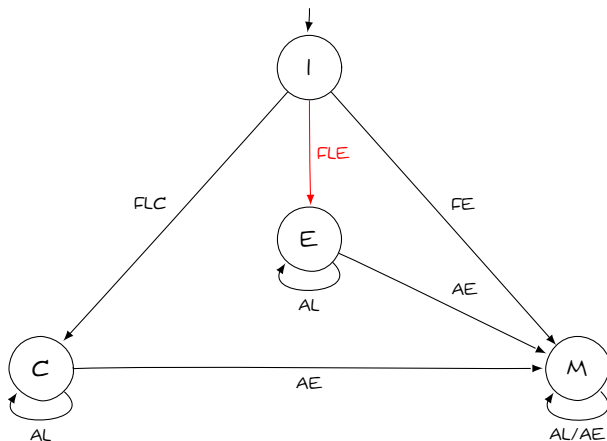
- Coerência de cache
  - Campos de metadados



Acerto de escrita (AE)

# Multiprocessamento

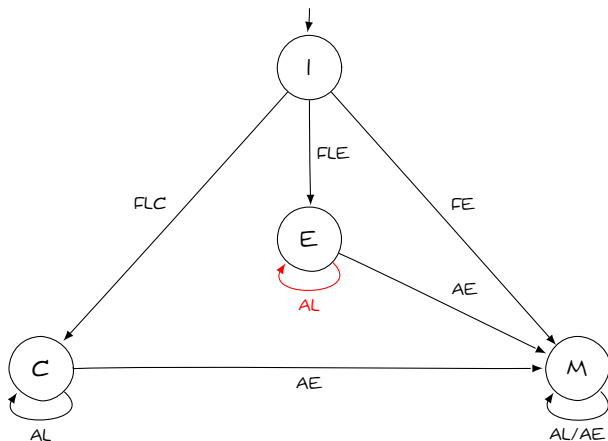
- ▶ Coerência de cache
  - ▶ Campos de metadados



Falta de leitura exclusiva (FLE)

# Multiprocessamento

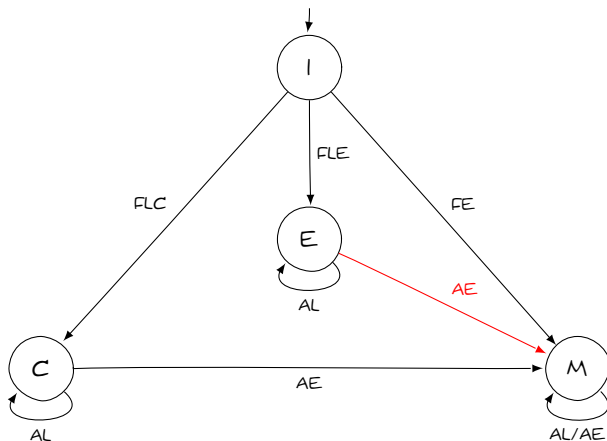
- Coerência de cache
  - Campos de metadados



Acerto de leitura (AL)

# Multiprocessamento

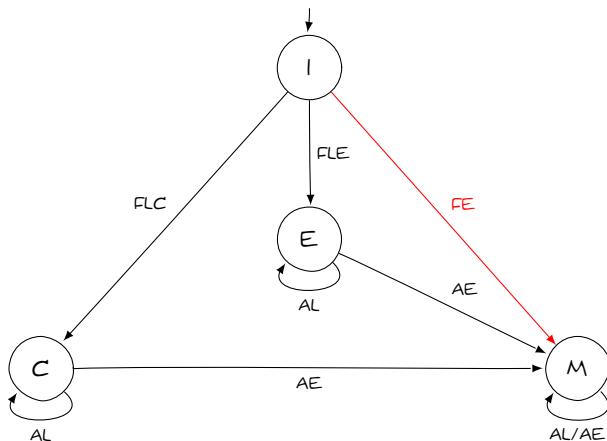
- Coerência de cache
  - Campos de metadados



Acerto de escrita (AE)

# Multiprocessamento

- Coerência de cache
  - Campos de metadados

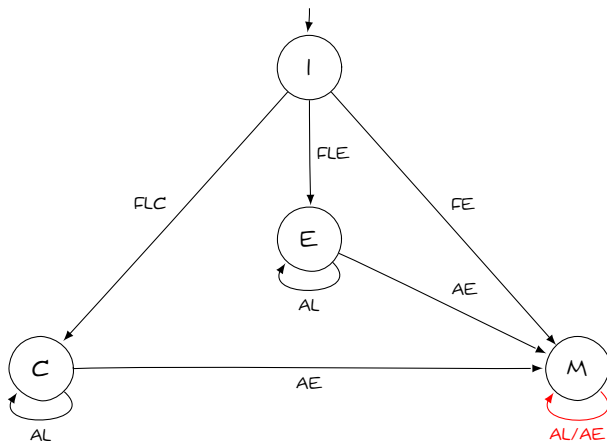


Falta de escrita (FE)



# Multiprocessamento

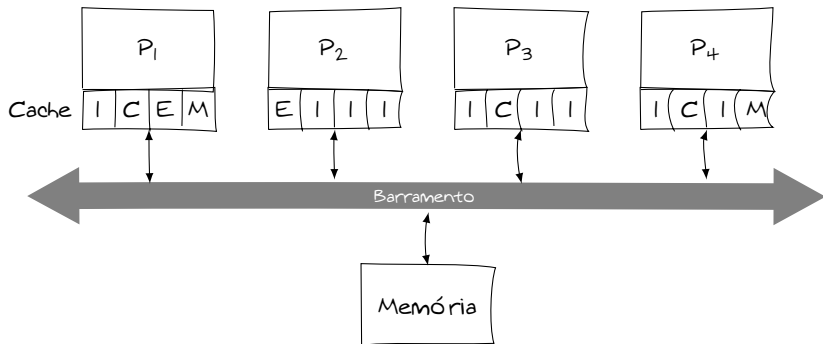
- Coerência de cache
  - Campos de metadados



Acerto de leitura (AL) ou de escrita (AE)

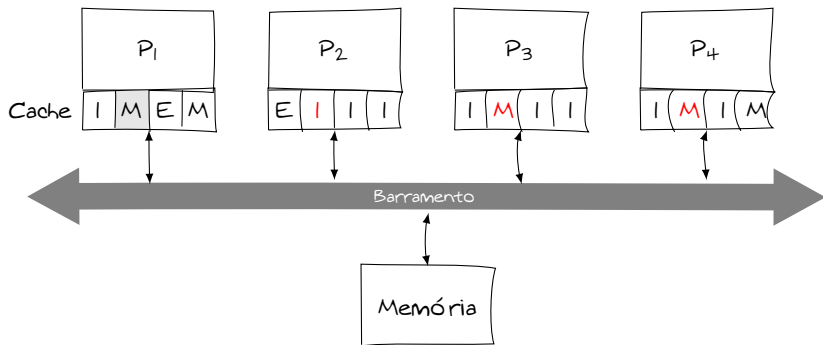
# Multiprocessamento

- ▶ Coerência de cache
  - ▶ Protocolo *snoopy*
    - ▶ Cada cache é responsável por armazenar os metadados das linhas de dados através de uma organização distribuída por mensagens



# Multiprocessamento

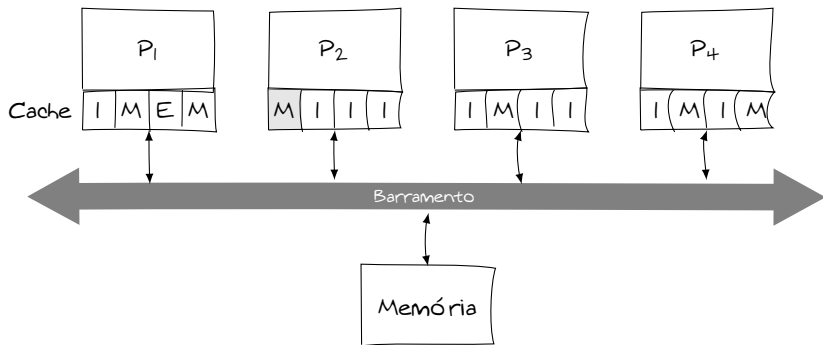
- ▶ Coerência de cache
  - ▶ Protocolo *snoopy*
    - ▶ Cada cache é responsável por armazenar os metadados das linhas de dados através de uma organização distribuída por mensagens



Modificação de bloco compartilhado

# Multiprocessamento

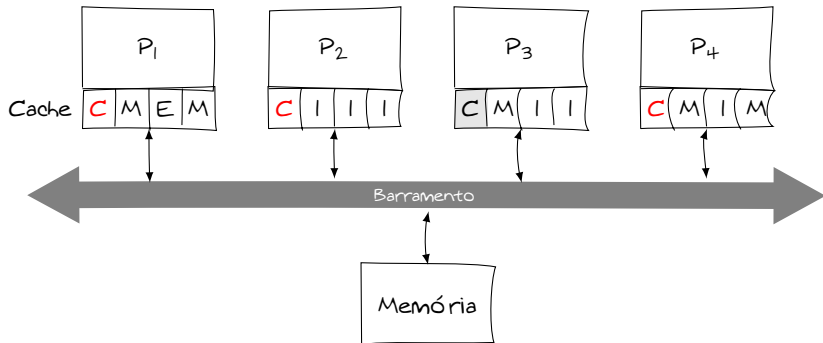
- ▶ Coerência de cache
  - ▶ Protocolo *snoopy*
    - ▶ Cada cache é responsável por armazenar os metadados das linhas de dados através de uma organização distribuída por mensagens



Modificação de bloco exclusivo

# Multiprocessamento

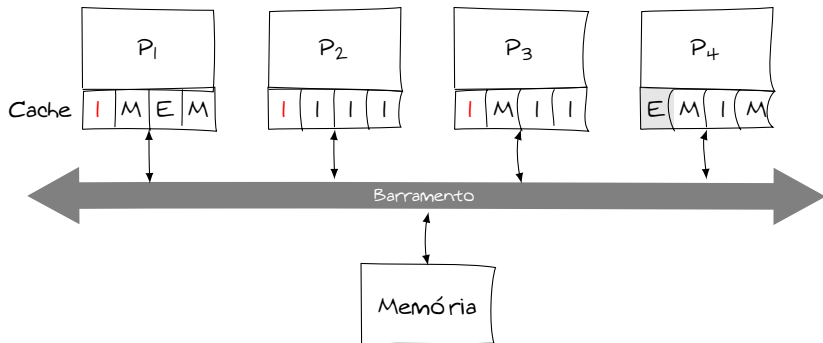
- ▶ Coerência de cache
  - ▶ Protocolo *snoopy*
    - ▶ Cada cache é responsável por armazenar os metadados das linhas de dados através de uma organização distribuída por mensagens



Compartilhamento de bloco modificado

# Multiprocessamento

- ▶ Coerência de cache
  - ▶ Protocolo *snoopy*
    - ▶ Cada cache é responsável por armazenar os metadados das linhas de dados através de uma organização distribuída por mensagens



Leitura exclusiva de bloco

# Multiprocessamento

- ▶ Coerência de cache
  - ▶ Protocolo *snoopy*
    - ▶ Organização distribuída x Robustez contra falhas

# Multiprocessamento

- ▶ Coerência de cache
  - ▶ Protocolo *snoopy*
    - ▶ Organização distribuída x Robustez contra falhas
    - ▶ Todas as operações são sincronizadas com envio de mensagens para todos os componentes

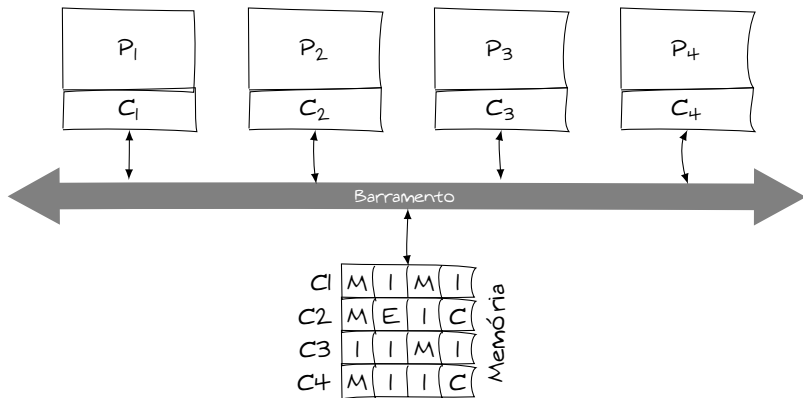


# Multiprocessamento

- ▶ Coerência de cache
  - ▶ Protocolo *snoopy*
    - ▶ Organização distribuída x Robustez contra falhas
    - ▶ Todas as operações são sincronizadas com envio de mensagens para todos os componentes
    - ▶ Não é escalável, uma vez que quanto mais processadores, maior é a quantidade de mensagens de controle que precisam ser transmitidas

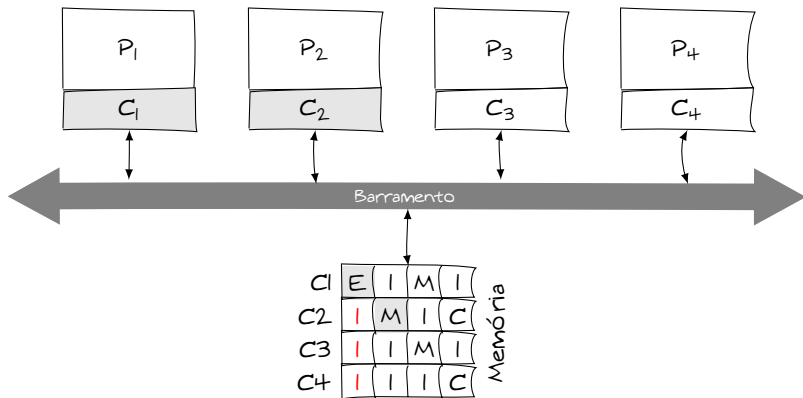
# Multiprocessamento

- ▶ Coerência de cache
  - ▶ Protocolo de diretório
    - ▶ Uma estrutura chamada diretório coleta e armazena as informações de acesso a memória, mantendo os metadados sobre as linhas de dados das caches



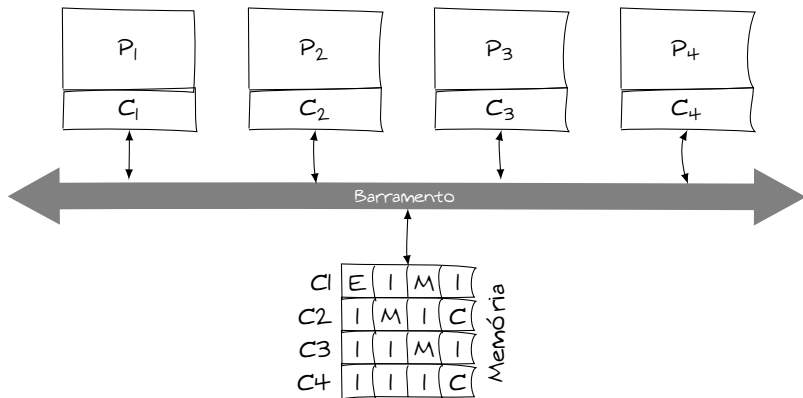
# Multiprocessamento

- ▶ Coerência de cache
  - ▶ Protocolo de diretório
    - ▶ Uma estrutura chamada diretório coleta e armazena as informações de acesso a memória, mantendo os metadados sobre as linhas de dados das caches



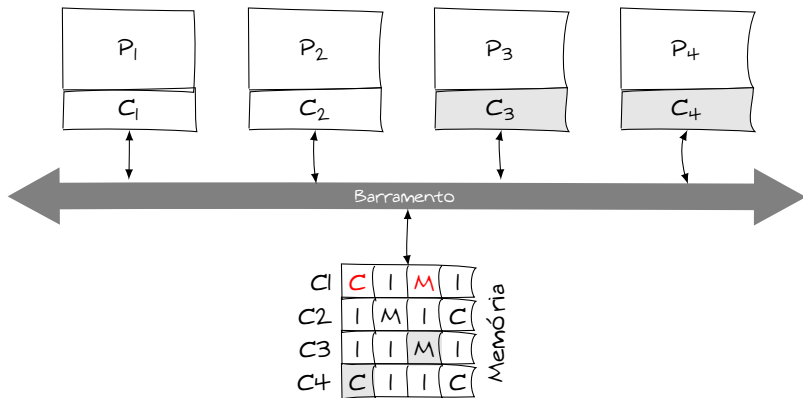
# Multiprocessamento

- ▶ Coerência de cache
  - ▶ Protocolo de diretório
    - ▶ Uma estrutura chamada diretório coleta e armazena as informações de acesso a memória, mantendo os metadados sobre as linhas de dados das caches



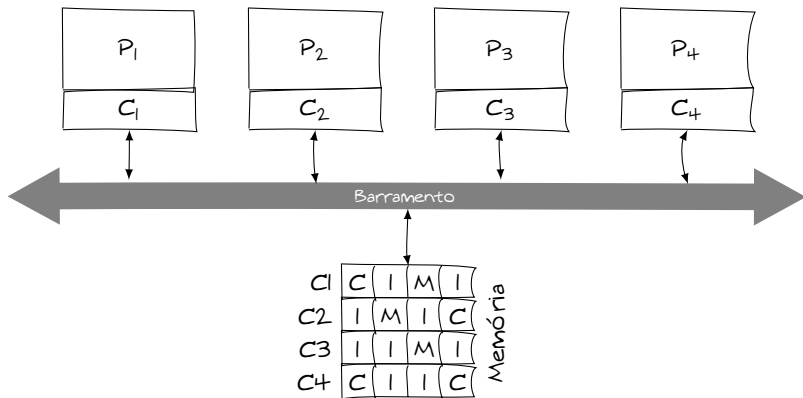
# Multiprocessamento

- ▶ Coerência de cache
  - ▶ Protocolo de diretório
    - ▶ Uma estrutura chamada diretório coleta e armazena as informações de acesso a memória, mantendo os metadados sobre as linhas de dados das caches



# Multiprocessamento

- ▶ Coerência de cache
  - ▶ Protocolo de diretório
    - ▶ Uma estrutura chamada diretório coleta e armazena as informações de acesso a memória, mantendo os metadados sobre as linhas de dados das caches



# Multiprocessamento

- ▶ Coerência de cache
  - ▶ Protocolo de diretório
    - ▶ Organização centralizada x Falha do dispositivo

# Multiprocessamento

- ▶ Coerência de cache
  - ▶ Protocolo de diretório
    - ▶ Organização centralizada x Falha do dispositivo
    - ▶ Não demanda envio de mensagens para todos os componentes do barramento para sincronização



# Multiprocessamento

- ▶ Coerência de cache
  - ▶ Protocolo de diretório
    - ▶ Organização centralizada x Falha do dispositivo
    - ▶ Não demanda envio de mensagens para todos os componentes do barramento para sincronização
    - ▶ Por este motivo é escalável, sendo adequado para plataformas com grande número de processadores

# Multiprocessamento

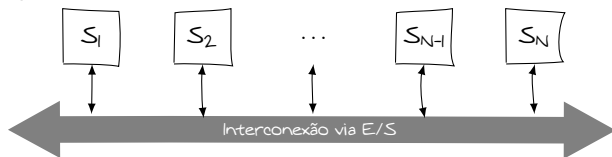
- ▶ Multiprocessamento assimétrico (AMP)
  - ▶ Esta classe de multiprocessamento oferece uma alternativa de alto desempenho e escalabilidade em comparação a abordagem simétrica

# Multiprocessamento

- ▶ Multiprocessamento assimétrico (AMP)
  - ▶ Esta classe de multiprocessamento oferece uma alternativa de alto desempenho e escalabilidade em comparação a abordagem simétrica
  - ▶ Os *clusters* são um dos exemplos mais representativos deste paradigma, onde um grupo de computadores interconectados por interfaces de entrada e saída cooperam como se fossem um único sistema

# Multiprocessamento

- ▶ Multiprocessamento assimétrico (AMP)
  - ▶ Esta classe de multiprocessamento oferece uma alternativa de alto desempenho e escalabilidade em comparação a abordagem simétrica
  - ▶ Os *clusters* são um dos exemplos mais representativos deste paradigma, onde um grupo de computadores interconectados por interfaces de entrada e saída cooperam como se fossem um único sistema



Alta disponibilidade

+

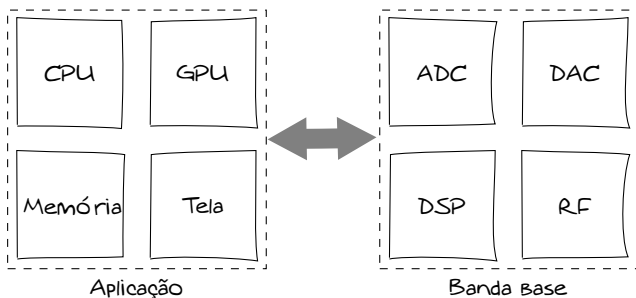
Escalabilidade

+

Relação custo  $\times$  desempenho

# Multiprocessamento

- ▶ Multiprocessamento assimétrico (AMP)
  - ▶ Os processadores do sistema possuem papéis distintos e com organizações de memória geralmente isoladas



# Multiprocessamento

- ▶ Multiprocessamento assimétrico (AMP)
  - ▶ Arquitetura híbrida de processadores com mesmo repertório de instruções, mas com diferentes níveis de desempenho e de consumo de potência

