
Universidade Federal do Maranhão
Centro de Ciências Exatas
Arquitetura de Computadores

# OpenMP

Otávio Belfort
Anderson Flávio

São Luís, 15 de Abril de 2021

# Sumário

# Sumário




[1] Open Multi-Processing

- Na década de 90. Desenvolveram extensões do compilador Fortran (instruções denominadas: **diretivas de execução**), que permitiam:
  - I

[1] Open Multi-Processing

- Na década de 90. Desenvolveram extensões do compilador Fortran (instruções denominadas: **diretivas de execução**), que permitiam:
  - Usuário, adicionar instruções para identificar “loops” que poderiam ser paralelizados;

[1] Open Multi-Processing

- Na década de 90. Desenvolveram extensões do compilador Fortran (instruções denominadas: **diretivas de execução**), que permitiam:
  - ıUsuário, adicionar instruções para identificar “loops” que poderiam ser paralelizados;O compilador passa a ser responsável pela paralelização automática desses “loop”.

[1] Open Multi-Processing

- Na década de 90. Desenvolveram extensões do compilador Fortran (instruções denominadas: **diretivas de execução**), que permitiam:
  - ıUsuário, adicionar instruções para identificar “loops” que poderiam ser paralelizados;O compilador passa a ser responsável pela paralelização automática desses “loop”.
- O primeiro padrão, ANSI X3H5, para testes foi lançado em 1994.

[1] Open Multi-Processing




- Na década de 90. Desenvolveram extensões do compilador Fortran (instruções denominadas: **diretivas de execução**), que permitiam:
  - ıUsuário, adicionar instruções para identificar “loops” que poderiam ser paralelizados;O compilador passa a ser responsável pela paralelização automática desses “loop”.
- O primeiro padrão, ANSI X3H5, para testes foi lançado em 1994.
- Um novo padrão foi desenvolvido em 1997 a partir do padrão anterior.

[1] Open Multi-Processing

- Na década de 90. Desenvolveram extensões do compilador Fortran (instruções denominadas: **diretivas de execução**), que permitiam:
  - ıUsuário, adicionar instruções para identificar “loops” que poderiam ser paralelizados;O compilador passa a ser responsável pela paralelização automática desses “loop”.
- O primeiro padrão, ANSI X3H5, para testes foi lançado em 1994.
- Um novo padrão foi desenvolvido em 1997 a partir do padrão anterior.
- Em 28 de Outubro de 1997 foi divulgado e disponibilizado o OpenMP Fortran API e no final de 1998 foi disponibilizado o OpenMP[1] C/C++ API

[1] Open Multi-Processing


ENGENHARIA DA COMPUTAÇÃO
UNIVERSIDADE FEDERAL DO MARANHÃO

- é uma especificação para um conjunto de ***diretivas do compilador***, ***rotinas de biblioteca*** e ***variáveis de ambiente*** que podem ser usadas para especificar paralelismo de alto nível nos programas Fortran e C / C ++. (OPENMP, 2019)

- é uma especificação para um conjunto de ***diretivas do compilador***, ***rotinas de biblioteca*** e ***variáveis de ambiente*** que podem ser usadas para especificar paralelismo de alto nível nos programas Fortran e C / C ++. (OPENMP, 2019)
- é uma interface de programação (API);

# Conceitos
## OpenMP



- é uma especificação para um conjunto de ***diretivas do compilador***, ***rotinas de biblioteca*** e ***variáveis de ambiente*** que podem ser usadas para especificar paralelismo de alto nível nos programas Fortran e C / C ++. (OPENMP, 2019)
- é uma interface de programação (API);
- portável;

# Conceitos
## OpenMP



- é uma especificação para um conjunto de ***diretivas do compilador***, ***rotinas de biblioteca*** e ***variáveis de ambiente*** que podem ser usadas para especificar paralelismo de alto nível nos programas Fortran e C / C ++. (OPENMP, 2019)
- é uma interface de programação (API);
- portável;
- baseada no modelo de programação paralela de memória compartilhada para arquiteturas de múltiplos processadores.

# Arquitetura
*Distributed Memory*

# Arquitetura
*Distributed Memory*



- Ambiente com vários processadores;

- Ambiente com vários processadores;
- Cada um com sua própria memória e interconectados por uma rede de comunicação.

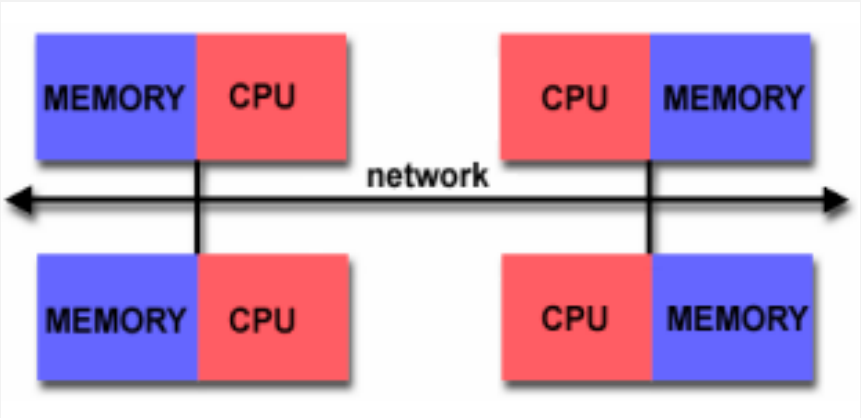


Figura: Distributed Memory

# Arquitetura
*Shared Memory*

- Ambiente com vários processadores;

- Ambiente com vários processadores;
- Compartilham o espaço de endereçamento de uma única memória.

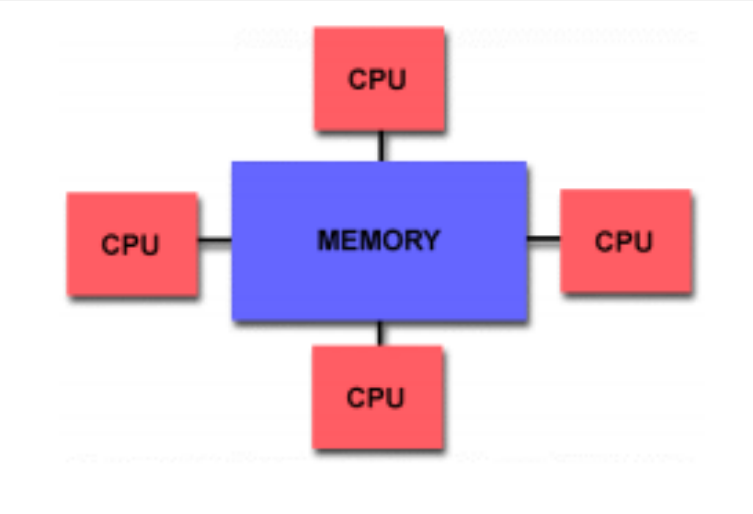


Figura: Distributed Memory


ENGENHARIA DA COMPUTAÇÃO
UNIVERSIDADE FEDERAL DO MARANHÃO

- Considera o número de instruções executadas em paralelo e o conjunto de dados sob os quais as instruções são submetidas.

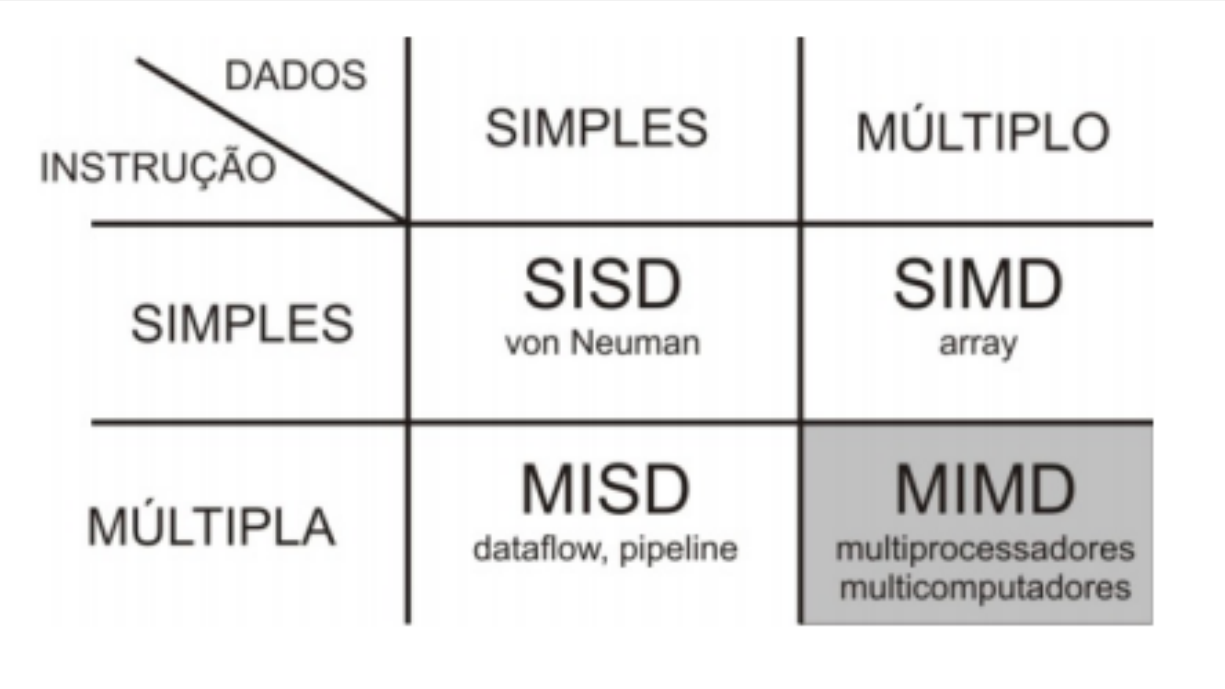


Figura: Classificação das arquiteturas de computadores segundo Flynn

# Arquitetura
## *MIMD*



[2] Uniform Memory Access

[3] Non-Uniform Memory Access

- **UMA**[2]: Tempo de acesso na memória é o mesmo.

[2] Uniform Memory Access
[3] Non-Uniform Memory Access

- **UMA**[2]: Tempo de acesso na memória é o mesmo.
- **NUMA**[3]: Tempo de acesso na memória depende da posição.

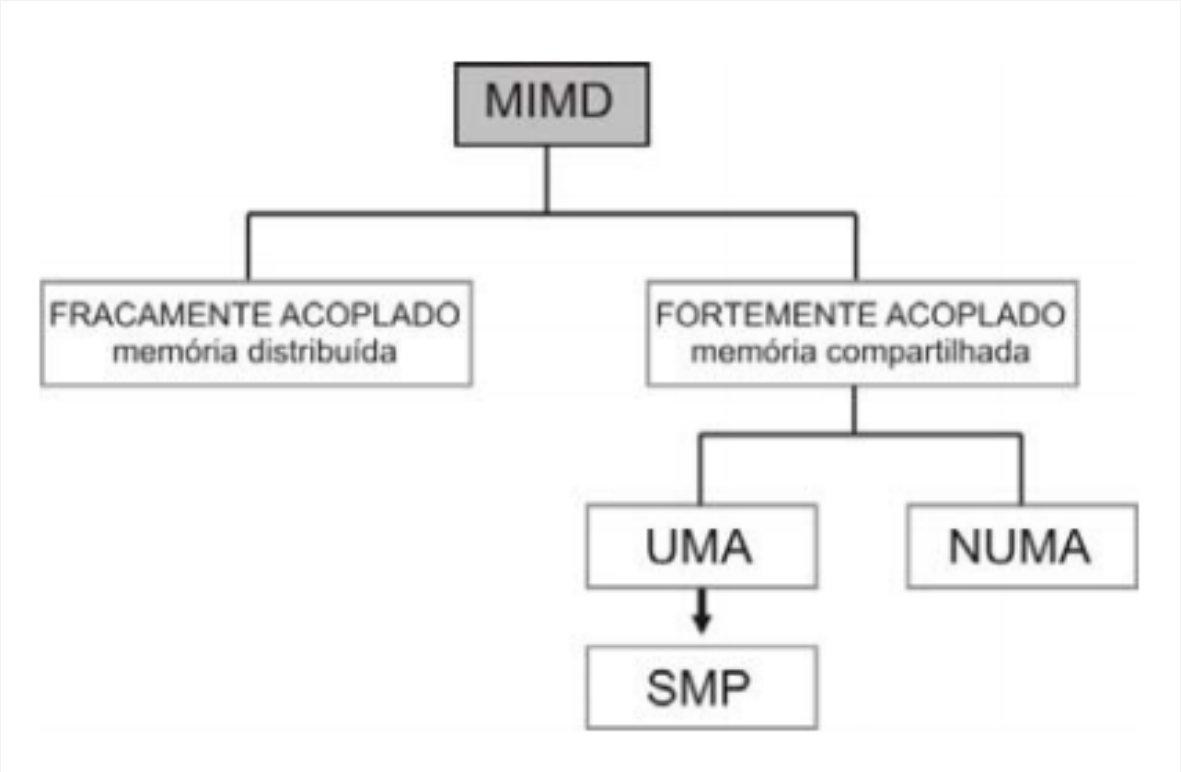


Figura: Classificação das arquiteturas MIMD

[2]Uniform Memory Access
[3]Non-Uniform Memory Access

# Principais Conceitos
## Threads

# Principais Conceitos
## Threads



- Uma thread é um processo “peso leve”.

# Principais Conceitos
## Threads



- Uma thread é um processo “peso leve”.
- Cada thread pode ser seu próprio fluxo de controle em um programa.

# Principais Conceitos
## Threads



- Uma thread é um processo “peso leve”.
- Cada thread pode ser seu próprio fluxo de controle em um programa.
- As threads podem compartilhar dados com outras threads, mas também têm dados privados.

# Principais Conceitos
## Threads



- Uma thread é um processo “peso leve”.
- Cada thread pode ser seu próprio fluxo de controle em um programa.
- As threads podem compartilhar dados com outras threads, mas também têm dados privados.
- As threads se comunicam através de uma área de dados compartilhada.

# Principais Conceitos
## Threads



- Uma thread é um processo “peso leve”.
- Cada thread pode ser seu próprio fluxo de controle em um programa.
- As threads podem compartilhar dados com outras threads, mas também têm dados privados.
- As threads se comunicam através de uma área de dados compartilhada.
- Uma equipe de threads é um conjunto de threads que cooperam em uma tarefa.

# Principais Conceitos
## Threads



- Uma thread é um processo “peso leve”.
- Cada thread pode ser seu próprio fluxo de controle em um programa.
- As threads podem compartilhar dados com outras threads, mas também têm dados privados.
- As threads se comunicam através de uma área de dados compartilhada.
- Uma equipe de threads é um conjunto de threads que cooperam em uma tarefa.
- A ***thread master*** é responsável pela coordenação da equipe de threads.

# Sumário


ENGENHARIA DA COMPUTAÇÃO
UNIVERSIDADE FEDERAL DO MARANHÃO




- A paralelização é explicitamente realizada com múltiplas *threads*.

- A paralelização é explicitamente realizada com múltiplas *threads*.
- A criação de *threads* é uma forma de um processo dividir a si mesmo em duas ou mais tarefas que podem ser executadas simultaneamente.

- A paralelização é explicitamente realizada com múltiplas *threads*.
- A criação de *threads* é uma forma de um processo dividir a si mesmo em duas ou mais tarefas que podem ser executadas simultaneamente.
- Cada *thread* possui sua própria pilha de execução.




- A paralelização é explicitamente realizada com múltiplas *threads*.
- A criação de *threads* é uma forma de um processo dividir a si mesmo em duas ou mais tarefas que podem ser executadas simultaneamente.
- Cada *thread* possui sua própria pilha de execução.
- Compartilha o mesmo endereço de memória com as outras *threads* do mesmo processo.




- A paralelização é explicitamente realizada com múltiplas *threads*.
- A criação de *threads* é uma forma de um processo dividir a si mesmo em duas ou mais tarefas que podem ser executadas simultaneamente.
- Cada *thread* possui sua própria pilha de execução.
- Compartilha o mesmo endereço de memória com as outras *threads* do mesmo processo.
- Cada processo possui seu próprio espaço de memória.

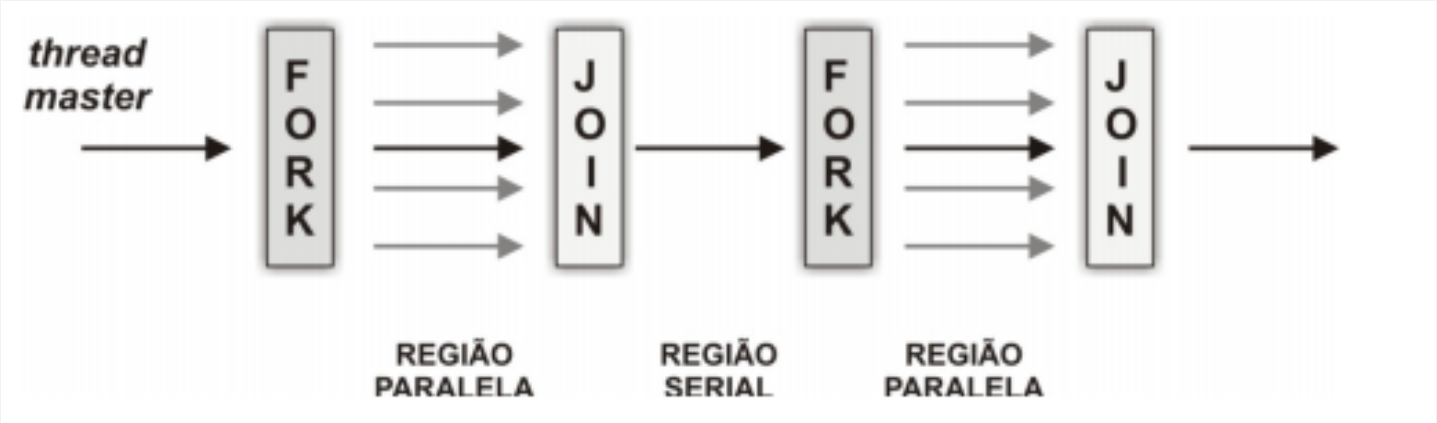


Figura: Modelo de programação do OpenMP


ENGENHARIA DA COMPUTAÇÃO
UNIVERSIDADE FEDERAL DO MARANHÃO

1. Variáveis de ambiente:
   - **OMP_NOME**: *OMP_NUM_THREADS*

1. Variáveis de ambiente:
   - **OMP_NOME**: *OMP_NUM_THREADS*
2. Diretivas de compilação:
   - **#pragma omp *diretiva [cláusula]***: *#pragma omp parallel*

1. Variáveis de ambiente:
   - **OMP_NOME**: *OMP_NUM_THREADS*
2. Diretivas de compilação:
   - **#pragma omp *diretiva [cláusula]***: *#pragma omp parallel*
3. Bibliotecas de serviço:
   - **omp_serviço (...)**: *omp_get_num_threads*

# Sumário

# OpenMP
## Vantagens

1. Facilidade de converção de programas sequencias em paralelos;




1. Facilidade de converção de programas sequencias em paralelos;
2. Maneira simples de explorar o paralelismo;

# OpenMP
## Vantagens



1. Facilidade de converção de programas sequencias em paralelos;
2. Maneira simples de explorar o paralelismo;
3. Fácil compreensão e uso das diretivas;

1. Facilidade de converção de programas sequencias em paralelos;
2. Maneira simples de explorar o paralelismo;
3. Fácil compreensão e uso das diretivas;
4. Possibilita o ajuste dinâmico do número de threads;




1. Facilidade de converção de programas sequencias em paralelos;
2. Maneira simples de explorar o paralelismo;
3. Fácil compreensão e uso das diretivas;
4. Possibilita o ajuste dinâmico do número de threads;
5. Compila e executa em ambientes paralelos e sequencial;

# OpenMP
## Vantagens



1. Facilidade de converção de programas sequencias em paralelos;
2. Maneira simples de explorar o paralelismo;
3. Fácil compreensão e uso das diretivas;
4. Possibilita o ajuste dinâmico do número de threads;
5. Compila e executa em ambientes paralelos e sequencial;
6. Possui uma robusta estrutura para suporte a programação paralela.

# Aplicações
## Exemplos

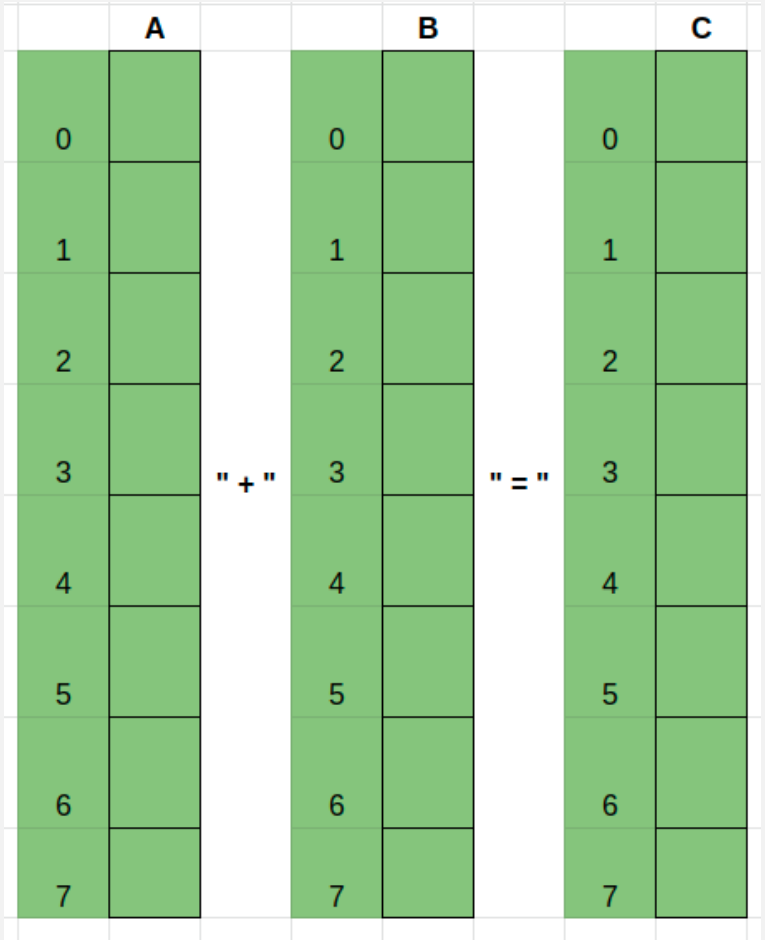


Figura: Soma de vetores

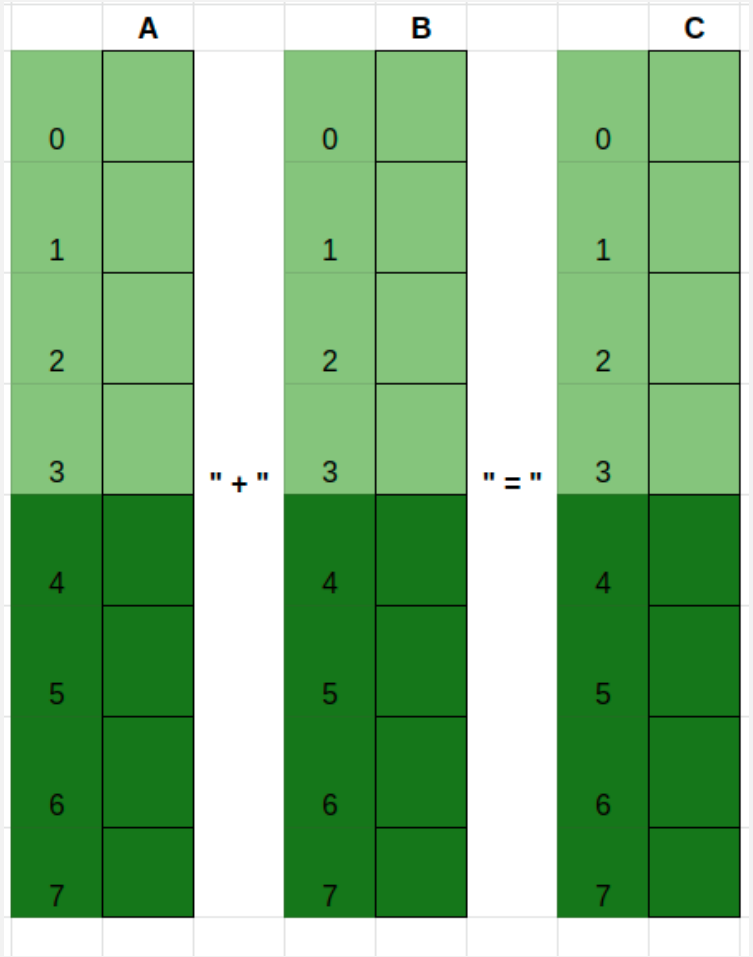


Figura: Soma de vetores em OpenMP - 2 Threads

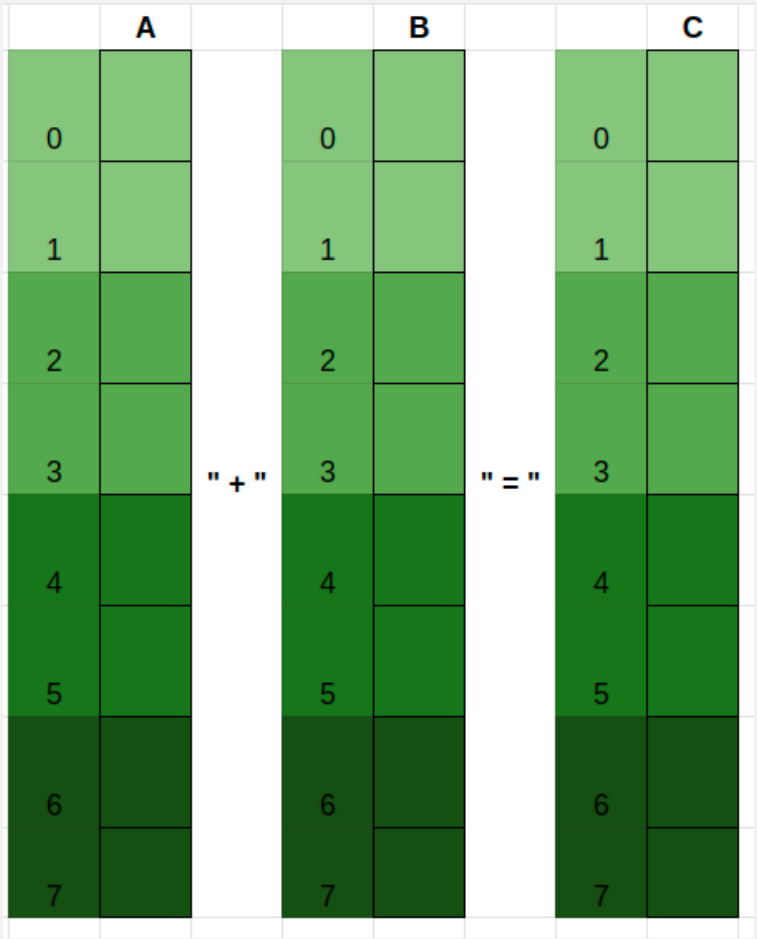


Figura: Soma de vetores em OpenMP - 4 Threads

# Aplicações
## Exemplos

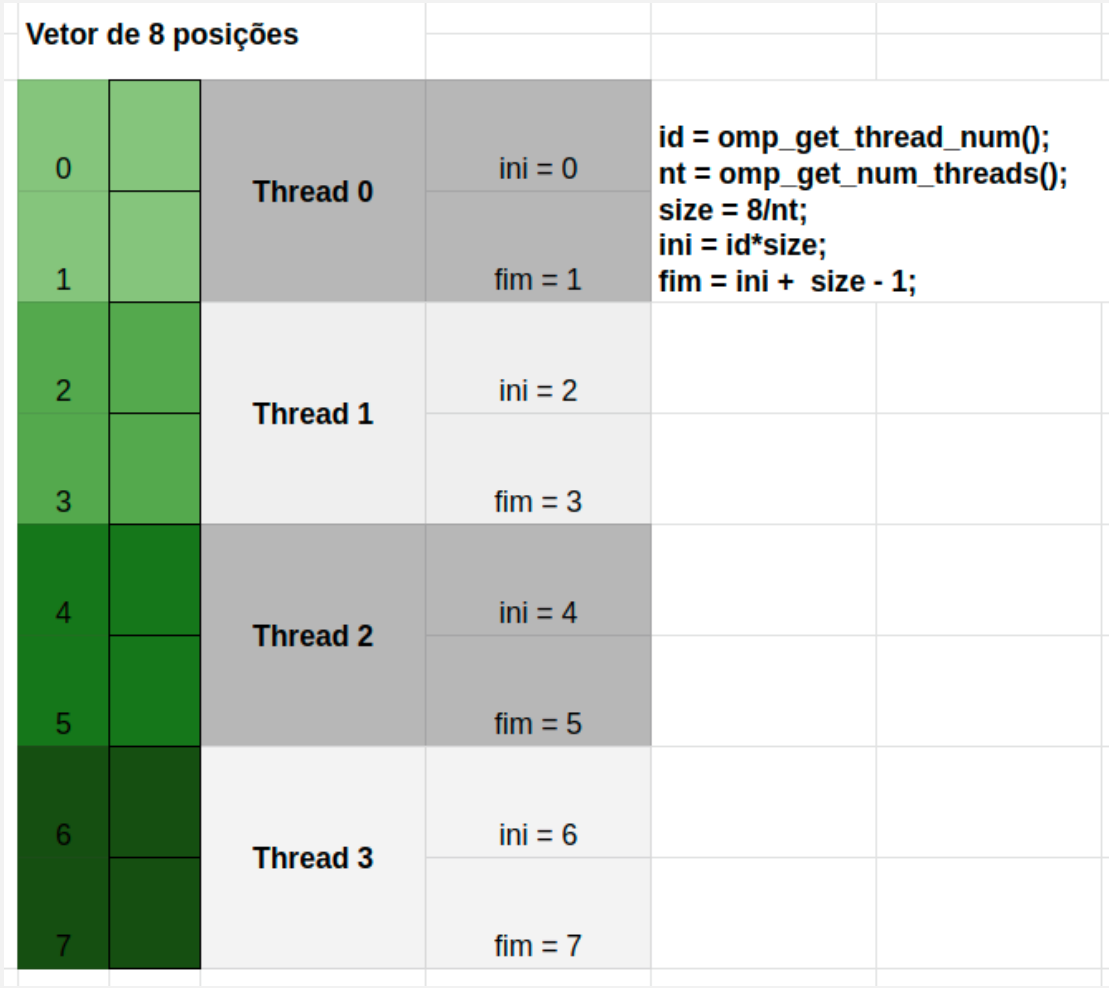


Figura: Funcionamento das *threads*

# Referências I




- http://www.inf.ufrgs.br/ afarah/files/openmp.pdf
- https://www.openmp.org/

**Obrigado!**

Otávio Belfort
Anderson Galvão
Link Github