
<https://youtu.be/HKaojhH2IG8>

Conceitos sobre Threads e OpenMP

Andreza Damasceno

Gabriel Leme

Gabriela Beato

Sumário

1 Conceitos básicos:

- O que é um processo;
- O que é uma thread;
- Diferença entre processo e thread.

2 Para que serve e onde usamos:

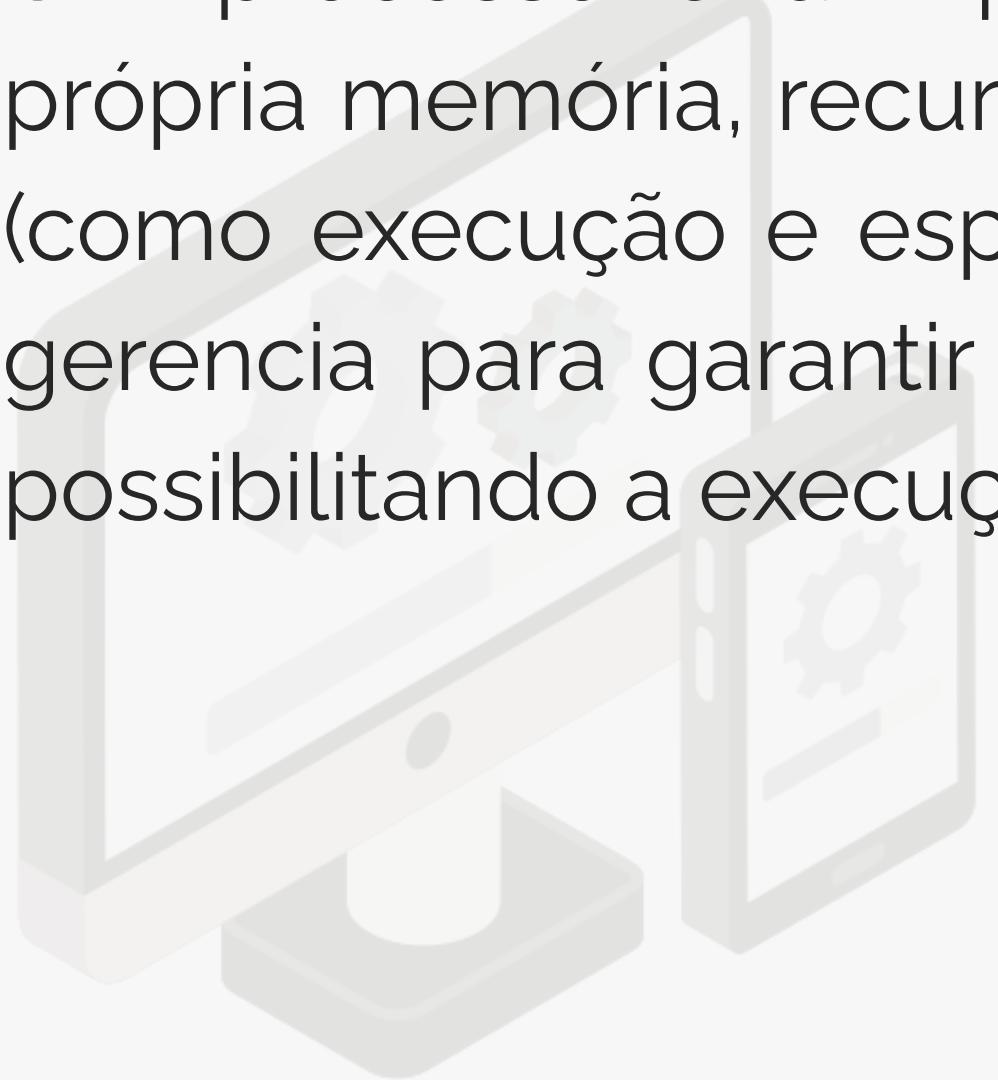
- Introdução ao Paralelismo e ao OpenMP;
- Threads: Função e Aplicações;
- Vantagens do Uso de Threads;
- OpenMP: Programação Paralela Simplificada.

3 OpenMP na prática:

- O que é o OpenMP;
- Conceitos básicos;
- Exemplo prático em código;
- Vantagens e cuidados.

O que é um Processo

Um processo é um programa em execução que tem sua própria memória, recursos e identificador. Ele muda de estado (como execução e espera) enquanto o sistema operacional o gerencia para garantir o uso eficiente e seguro dos recursos, possibilitando a execução simultânea de vários programas.



O que é uma Thread

Uma thread é a menor unidade de execução dentro de um processo, compartilhando memória e recursos com outras threads do mesmo processo, mas possuindo seu próprio fluxo de execução, permitindo que o programa realize múltiplas tarefas simultaneamente.



Diferença entre Processo e Threads

- Processos: Programas em execução com memória e recursos próprios, funcionando de forma independente. São mais pesados e isolados;
- Threads: Unidades de execução dentro de um processo, compartilham memória e recursos com outras threads, mas possuem fluxo de execução próprio. São mais leves e permitem multitarefa dentro de um mesmo programa.

Introdução ao Paralelismo e ao OpenMP

- O avanço dos processadores com múltiplos núcleos tornou o paralelismo essencial no desenvolvimento de softwares;
- O paralelismo divide tarefas em partes menores executadas simultaneamente, otimizando o desempenho;
- Segundo Perez *et al.* (2023), cresce a necessidade de profissionais capacitados em programação paralela;
- Ainda há defasagem no ensino dessa área (Almeida Jr. *et al.*, 2022);
- Compreender Threads e OpenMP é fundamental para acompanhar as demandas da computação moderna.

Threads: Função e Aplicações

- Threads permitem que partes de um programa sejam executadas concorrentemente, compartilhando a mesma memória;
- Segundo Karakaya e Mishra, um processo pode ter múltiplos caminhos de execução simultâneos;
- Usadas em:
 - Sistemas operacionais (controle de processos);
 - Servidores e navegadores web;
 - Jogos digitais e simulações;
 - Aplicações científicas e computação de alto desempenho.

Vantagens do Uso de Threads

- Conteúdo resumido (ao lado ou acima da imagem);
- Melhor aproveitamento dos núcleos do processador;
- Redução do tempo total de execução;
- Maior velocidade e eficiência em tarefas simultâneas;
- Permite respostas mais rápidas em aplicações interativas.

OpenMP: Programação Paralela Simplificada

- OpenMP (Open Multi-Processing) é uma API para C, C++ e Fortran;
- Facilita a implementação de paralelismo em memória compartilhada;
- Permite criar programas paralelos com diretivas simples de compilação;
- É multiplataforma e escalável, podendo rodar de PCs a supercomputadores;
- Simplifica o controle de threads e maximiza o desempenho.

O que é o OpenMP

- O OpenMP é uma API usada para criar programas que executam várias tarefas ao mesmo tempo (multithreading) em sistemas com memória compartilhada;
- Suporta **C**, **C++** e **Fortran** e funciona em **Windows** e **Linux**;
- O programador usa diretivas (#pragma omp) no código para indicar as partes que podem rodar em paralelo;
- O compilador é responsável por criar, gerenciar e sincronizar automaticamente as threads (as linhas de execução).

(PENHA, CORRÊA e MARTINS, 2002).

Para mais informações...



<https://www.openmp.org/specifications/>

Conceitos básicos

- **Thread:** Linha de execução de um programa;
- **Paralelismo:** Executar várias tarefas ao mesmo tempo;
- **Fork (Divisão):** O programa divide uma tarefa grande em **várias tarefas menores** que podem ser executadas concorrentemente (em paralelo). Cada subtarefa é independente, permitindo que múltiplas threads trabalhem ao mesmo tempo;
- **Join (Junção):** Depois que todas as subtarefas são concluídas, os resultados são reunidos (ou “juntados”) para produzir o resultado final da tarefa original.

Exemplo prático em código:

```
C openmp.c > ...
1 #include <stdio.h>
2 #include <omp.h>
3
4 int main() {
5     #pragma omp parallel
6     {
7         printf("Thread: %d\n", omp_get_thread_num());
8     }
9     return 0;
10 }
```

Terminal:

```
Andreza Damasceno@ MINGW64 /c/Users/Andreza
$ gcc -fopenmp openmp.c -o openmp.exe

Andreza Damasceno@ MINGW64 /c/Users/Andreza
$ ./openmp.exe
Thread: 9
Thread: 1
Thread: 0
Thread: 7
Thread: 5
Thread: 6
Thread: 4
Thread: 2
Thread: 8
Thread: 10
Thread: 3
Thread: 11
```

Vantagens e Cuidados

Vantagens:

- Fácil de usar: o programador não precisa gerenciar threads manualmente;
- Portátil e padronizado: funciona em várias plataformas;
- Melhor desempenho em tarefas que podem ser divididas, como processamento de imagem;
- Permite melhor aproveitamento do hardware.

Vantagens e Cuidados

Cuidados:

- O OpenMP não garante o uso mais eficiente da memória (não controla onde os dados ficam);
- Nem sempre traz o melhor tempo de resposta possível, dependendo do sistema.

Referências

- ALMEIDA JUNIOR, C. S.; FARIAS, L. A. F.; ARAÚJO, A. S. **Um Mapeamento Sistemático da Literatura sobre o Ensino e Aprendizagem da Programação Paralela com o uso de Placas Gráficas.** In: Anais do XXXII Simpósio Brasileiro de Informática na Educação (SBIE 2022). Porto Alegre: Sociedade Brasileira de Computação, 2022.
- GRAN CURSOS ONLINE. **Sistemas Operacionais - Gerenciamento de processos.** Blog Gran Cursos Online, 2024. Disponível em: <<https://blog.grancursosonline.com.br/sistemas-operacionais-gerenciamento-de-processos/>>. Acesso em: 09 out. 2025.
- IBM. **Entendendo threads e processos.** AIX 7.3.0 Documentation, 2025. Disponível em: <<https://www.ibm.com/docs/pt-br/aix/7.3.0?topic=programming-understanding-threads-processes>>. Acesso em: 11 out. 2025.
- KARAKAYA, Z.; MISHRA, A. **Teaching Parallel Computing Concepts Using Real-Life Applications.** In: Proceedings of the 2020 IEEE Frontiers in Education Conference (FIE). Uppsala: IEEE, 2020.
- OPENMP (introdução e diretiva parallel).** YouTube, 2020. 29:30. Publicado pelo canal Márcio Castro. Disponível em: <https://youtu.be/HY62wPtefAY?si=MLAI8PiudnyLz_vM>. Acesso em: 18 out. 2025.
- PENHA, Dulcinéia Oliveira da; CORRÊA, João Batista Torres; MARTINS, Carlos Augusto Paiva S. **Análise comparativa do uso de multi-thread e OpenMP aplicados a operações de convolução de imagem.** Anais do WSCAD. Belo Horizonte: Pontifícia Universidade Católica de Minas Gerais, Laboratório de Sistemas Digitais e Computacionais, Instituto de Informática, Programa de Pós-Graduação em Engenharia Elétrica, 2002. p. 118–125.

Referências

- PEREZ, J. R. F. et al. **Ensino de Programação Paralela na Educação Básica: Uma Revisão Sistemática da Literatura, Análise Bibliométrica e Agenda para Estudos Futuros.** In: Anais do XLIII Congresso da Sociedade Brasileira de Computação (CSBC 2023). Porto Alegre: Sociedade Brasileira de Computação, 2023.
- TEC NOBLOG. **O que são threads do processador e quais os benefícios do multithreading?** Blog Tecnoblog, 2024. Disponível em: <<https://tecnoblog.net/responde/o-que-e-thread-processador/>>. Acesso em: 11 out. 2025.

OBRIGADO

Andreza Damasceno; Gabriel Leme; Gabriela Beato.