

# Introdução a *Pthreads* e *OpenMP*

---

Pedro Bruel

*phrb@ime.usp.br*

# Introdução

---

# Slides

Os slides e todo o código fonte estão no **GitHub**:

- <https://phrb.github.io/PPD>

# Motivação



# Programação Paralela: Motivação

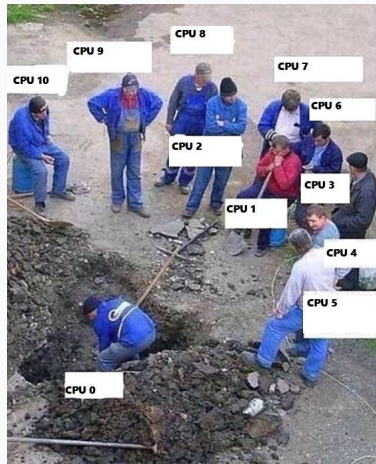
Por que usar programação paralela?

Desempenho:

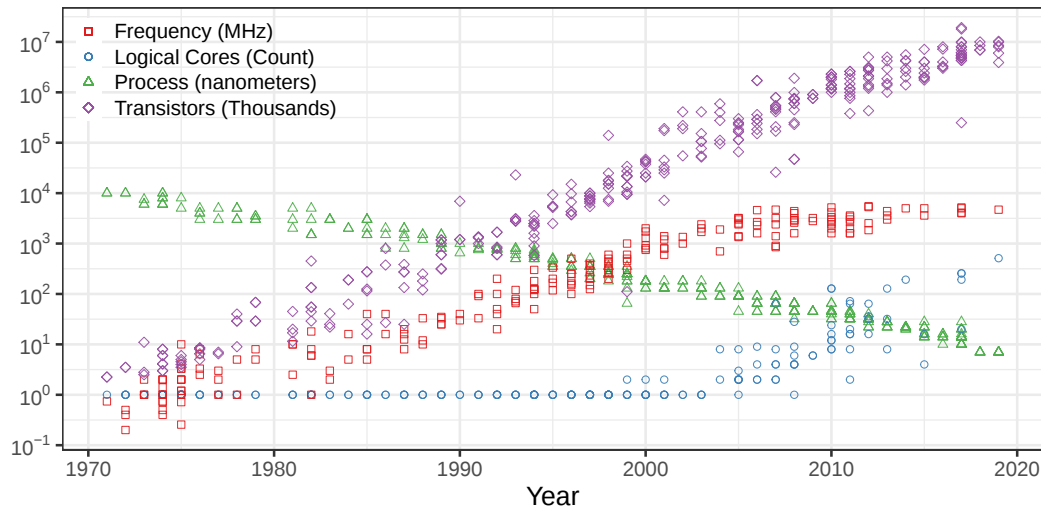
- Arquiteturas paralelas
- Memória Compartilhada
- SMP, hyperthreaded, multi-core, NUMA,  
...

Modelagem:

- Descrever paralelismo natural
- Tarefas independentes

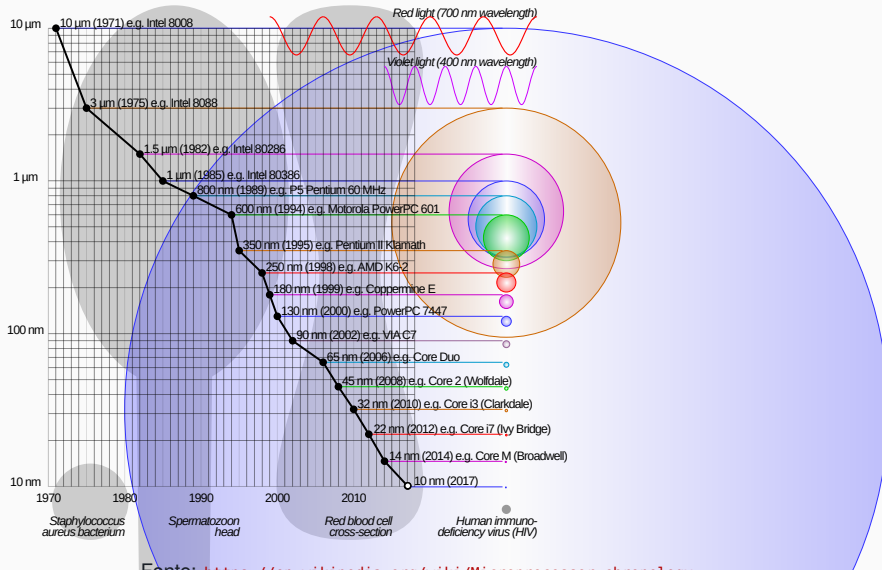


# 49 Anos de Tendências em Microprocessadores



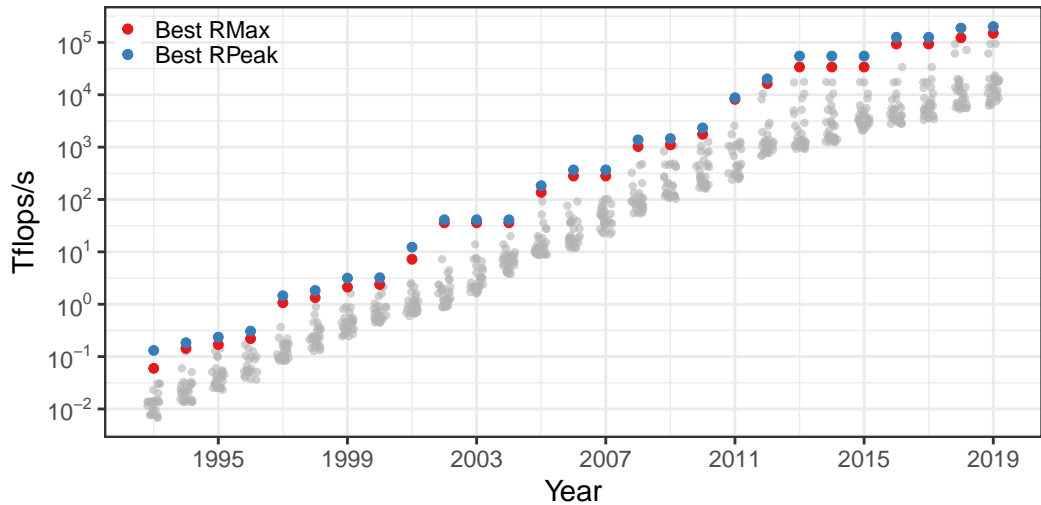
Fonte: [https://en.wikipedia.org/wiki/Transistor\\_count](https://en.wikipedia.org/wiki/Transistor_count), [https://en.wikipedia.org/wiki/Microprocessor\\_chronology](https://en.wikipedia.org/wiki/Microprocessor_chronology)

# Um Senso de Escala: Tamanho de Microprocessadores



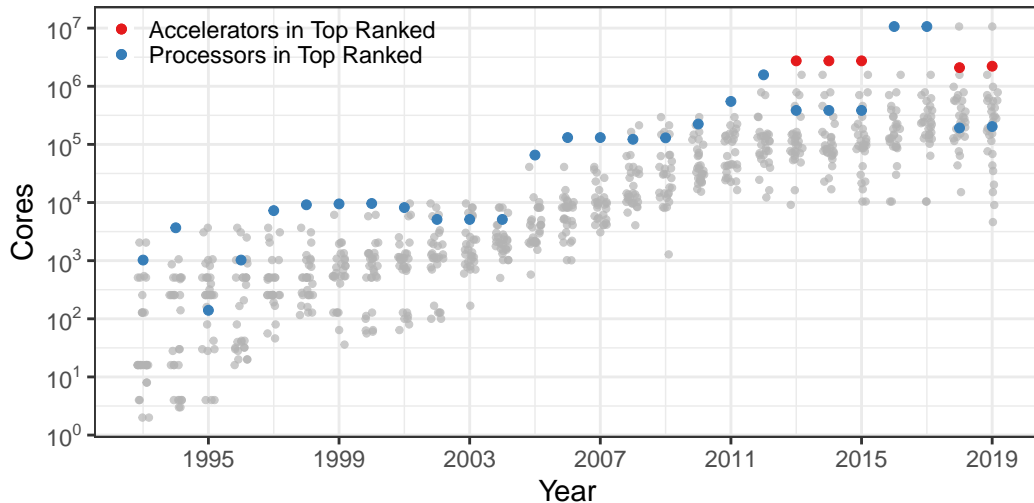
Fonte: [https://en.wikipedia.org/wiki/Microprocessor\\_chronology](https://en.wikipedia.org/wiki/Microprocessor_chronology)

## Top500: *RPeak* e *RMax*





## Top500: Núcleos de *processador* e *acelerador*



## IEEE POSIX Threads

---

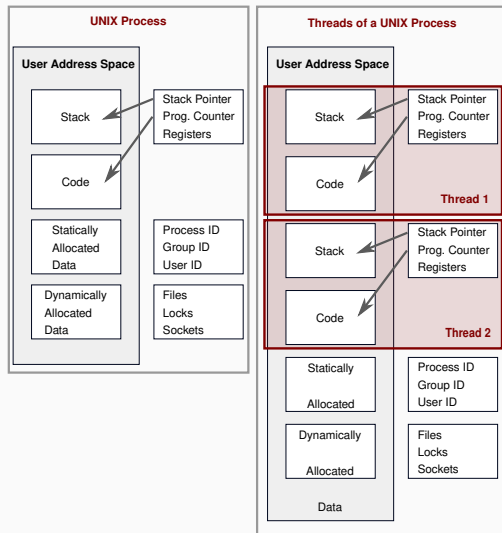
# IEEE POSIX Threads

## IEEE e POSIX:

- Institute of Electrical and Electronics Engineers (**IEEE**)
- Portable Operating System Interface (**POSIX**)

## IEEE POSIX Threads:

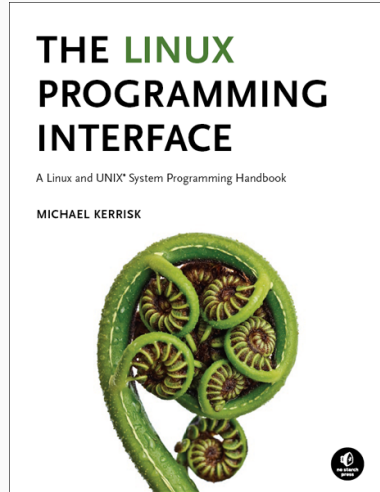
- Define um **modelo de execução**
- **Independente** de linguagens
- Execução paralela de “**fluxos de trabalho**” (**threads**)
- Define uma API para **criação e controle** de threads
- **Não define** detalhes de implementação



# Threads: The Linux Programming Interface

## The Linux Programming Interface (Michael Kerrisk, 2010)

- Processos e threads
- Capítulos 24 a 33
- Detalhes sobre o kernel



# API Pthreads

~100 funções prefixadas por  
pthread\_:

- Gerenciamento
- Mutexes
- Variáveis condicionais
- Sincronização

Prefixo	Funcionalidade
pthread_	Gerenciamento
pthread_attr_	Atributos
pthread_mutex_	Mutexes
pthread_mutexattr_	Atributos de Mutexes
pthread_cond_	Variáveis condicionais
pthread_condattr_	Atributos de condicionais
pthread_key_	Dados específicos de threads
pthread_rwlock_	<i>Locks</i> de leitura e escrita
pthread_barrier_	Barreiras e sincronização

**Tabela 1:** Algumas funções da API Pthreads

# Pthreads: Tutorial

## POSIX Threads Programming:

- Blaise Barney, Lawrence Livermore National Laboratory
- <https://hpc-tutorials.llnl.gov/posix/>

# POSIX Threads: Hello, World!

```
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#define NUM_THREADS 5
void *PrintHello(void *threadid) {
    long tid;
    tid = (long)threadid;
    printf("Hello World! It's me, thread #%ld!\n", tid);
    pthread_exit(NULL);
}
int main(int argc, char *argv[]) {
    pthread_t threads[NUM_THREADS];
    int rc;
    long t;
    for(t=0;t<NUM_THREADS;t++){
        printf("In main: creating thread %ld\n", t);
        rc = pthread_create(&threads[t], NULL, PrintHello, (void *)t);
        if(rc) {
            printf("ERROR; return code from pthread_create() is %d\n",
                rc);
            exit(-1);
        }
    }
    pthread_exit(NULL);
}
```

# POSIX Threads: Mais Exemplos

Exemplos de código:

[https://github.com/phrb/PPD/tree/main/lectures/tex/pthreads\\_omp/code\\_samples/pthreads](https://github.com/phrb/PPD/tree/main/lectures/tex/pthreads_omp/code_samples/pthreads)

- Hello, World!
- Argumentos
- *Join*
- Servidor IRC: <https://github.com/phrb/simple-irc-server>



# OpenMP



# OpenMP

Open Multi-Processing (OpenMP):

- API para paralelismo **multithreaded** e de **memória compartilhada**
- **Diretivas de compilador**
- Biblioteca de **Tempo de Execução (Runtime)**
- Variáveis de ambiente

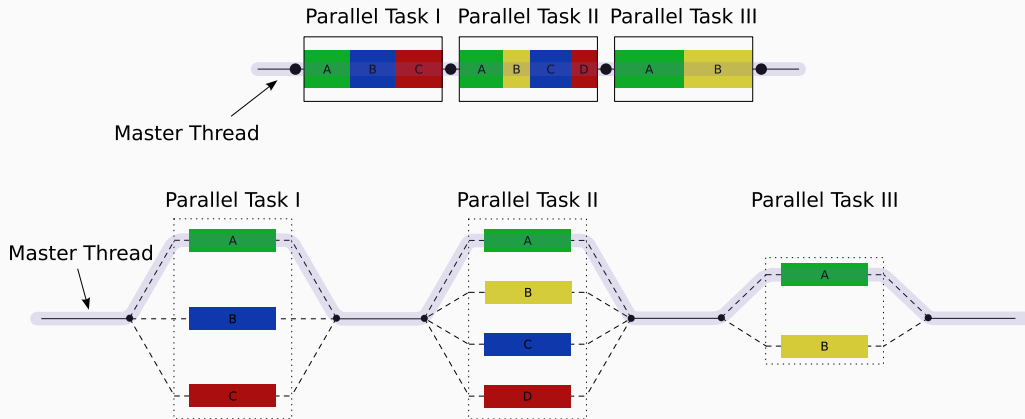
Objetivos:

- Padronizar
- Simplificar
- Promover **portabilidade**

# OpenMP: Modelo de Programação

- Threads **dinâmicas**
- Paralelismo **explícito e aninhável**
- **Diretivas** de compilador
- Modelo **Fork-Join**

# OpenMP: Fork-Join



# OpenMP: Diretivas

Usadas para:

- Criar regiões paralelas
- Distribuir blocos de código
- Distribuir iterações de laços
- Sincronizar threads
- ...

Modelo:

```
#pragma omp directive [clause, ...]
```

Exemplo:

```
#pragma omp parallel default(shared) private(beta,pi)
```

# OpenMP: Biblioteca Runtime

Usada para:

- Obter e definir **número de threads**
- Obter **IDs de threads**
- Obter **região paralela e nível de aninhamento**
- Obter, criar e destruir **locks**
- ...

Exemplo:

```
#include <omp.h>
int omp_get_num_threads(void)
```

# OpenMP: Variáveis de Ambiente

Usadas para:

- Definir número de threads
- Distribuir iterações de laços
- Associar threads a processadores
- Configurar paralelismo aninhado
- Configurar threads dinâmicas
- ...

Exemplo:

```
export OMP_NUM_THREADS=8
```

# OpenMP: Tutorial

## OpenMP Programming:

- Blaise Barney, Lawrence Livermore National Laboratory
- <https://computing.llnl.gov/tutorials/openMP>



# OMP: Hello, World!

```
#include <stdio.h>
#include <omp.h>

int main(int argc, char *argv[]){
    int nthreads, tid;

    #pragma omp parallel private(tid)
    {
        tid = omp_get_thread_num();
        printf("Hello World from thread = %d\n", tid);

        if(tid == 0){
            nthreads = omp_get_num_threads();
            printf("Number of threads = %d\n", nthreads);
        };
    };
    return 0;
};
```

# OpenMP: Mais Exemplos

Exemplos de código:

[https://github.com/phrb/presentations/tree/master/aula-pthreads/code\\_samples/omp](https://github.com/phrb/presentations/tree/master/aula-pthreads/code_samples/omp)

- Hello, World!
- Parallel for
- Reduction
- Critical section

# Introdução a *Pthreads* e *OpenMP*

---

Pedro Bruel

*phrb@ime.usp.br*