

# INTRODUÇÃO A *Pthreads* & *OpenMP*

---

Pedro Bruel

*phrb@ime.usp.br*

2020



Versão Quarentena

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

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

# PROGRAMAÇÃO CONCORRENTE: MOTIVAÇÃO

Por que usar programação concorrente?

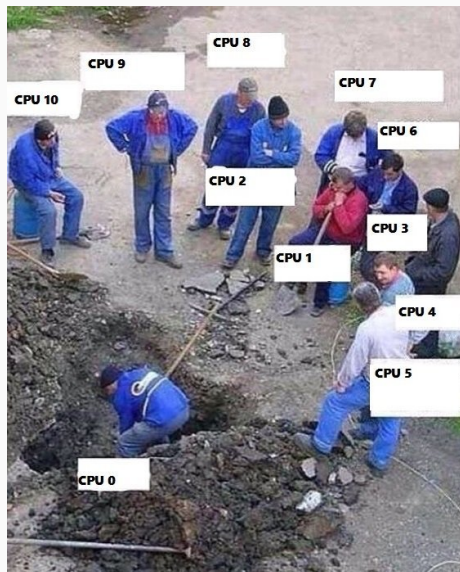
## Desempenho:

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

## Modelagem:

- Descrever paralelismo natural
- Tarefas independentes

# PROGRAMAÇÃO CONCORRENTE: MOTIVAÇÃO



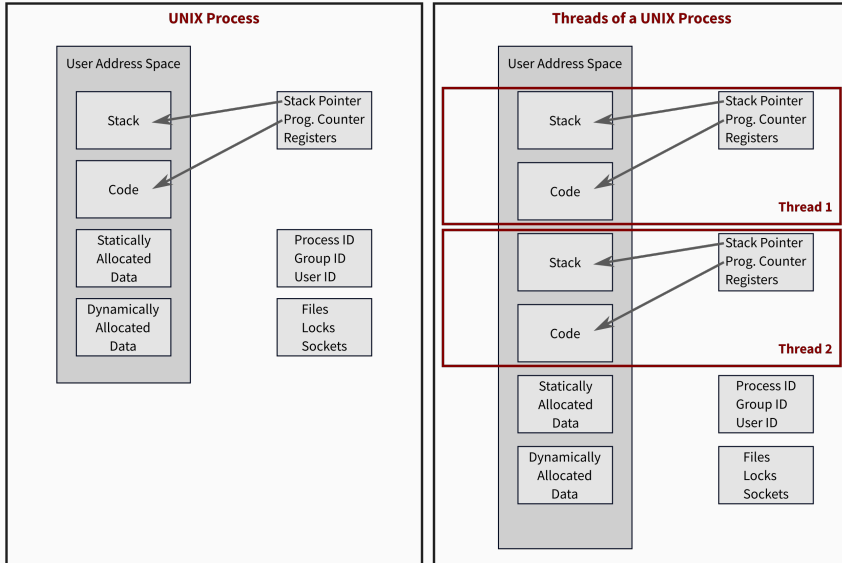
## IEEE & POSIX:

- 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

# PROCESSOS & THREADS



~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

Algumas funções da API Pthreads



## POSIX Threads Programming:

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

# POSIX THREADS: HELLO, WORLD!

```
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#define NUM_THREADS 5
void *print_hello(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 error_code;
    long t;
    for(t = 0; t < NUM_THREADS; t++){
        printf("In main: creating thread %ld\n", t);
        error_code = pthread_create(&threads[t], NULL,
                                   print_hello, (void *) t);

        if (error_code){
            printf("ERROR pthread_create(): %d\n", error_code);
            exit(-1);
        }
    };
    pthread_exit(NULL);
};
```

## Exemplos de código:

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

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

## Open Multi-Processing (OpenMP):

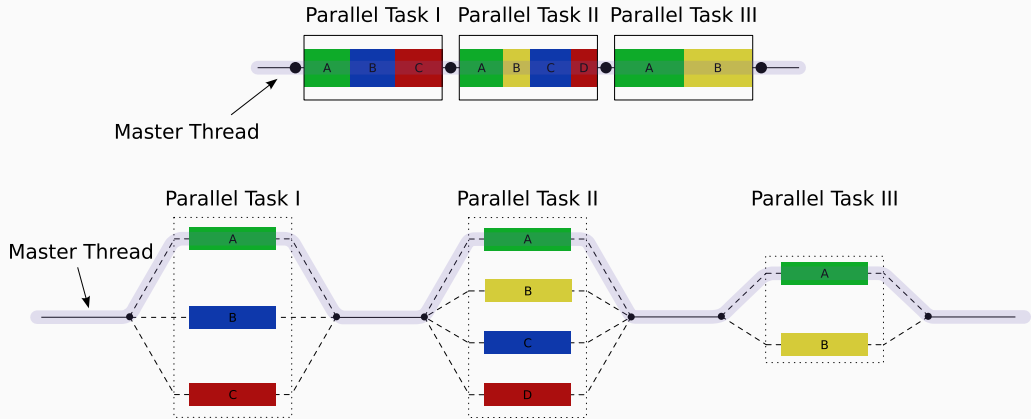
- API para paralelismo **multithreaded** e de **memória compartilhada**
- **Diretivas de compilador**
- Biblioteca de **Tempo de Execução (Runtime)**
- Variáveis de ambiente

## Objetivos:

- Padronizar
- Simplificar
- Promover **portabilidade**

- 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)
```

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)
```



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 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;
};
```

## 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* & *OpenMP*

---

Pedro Bruel

*phrb@ime.usp.br*

2020



Versão Quarentena