

INTRODUÇÃO A PTHREADS & OPENMP

Pedro Bruel

phrb@ime.usp.br

17 de abril de 2017



Instituto de Matemática e Estatística
Universidade de São Paulo

SOBRE O EP1

Na aula passada faltou:

- Data de Entrega do EP1: [25/04](#)
- Demonstrar GNU screen

ROTEIRO

1. Motivação
2. IEEE POSIX Threads
3. OpenMP

SLIDES



Os slides e todo o código fonte estão no [GitHub](#):

- github.com/phrb/aula-pthreads-openmp

PROGRAMAÇÃO CONCORRENTE: MOTIVAÇÃO

Por que usar programação concorrente?

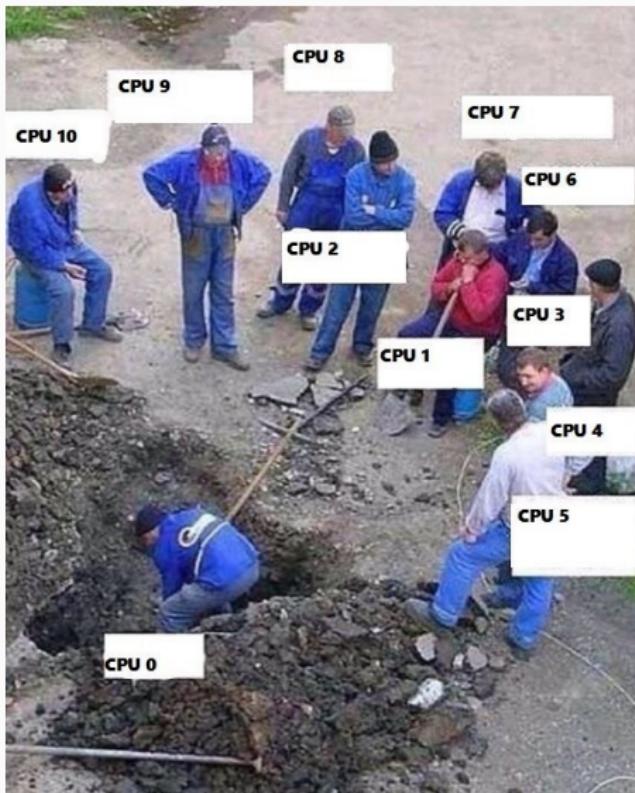
Desempenho:

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

Modelagem:

- Descrever paralelismo natural
- Tarefas independentes

PROGRAMAÇÃO CONCORRENTE: MOTIVAÇÃO



PROGRAMAÇÃO CONCORRENTE: MOTIVAÇÃO

IEEE POSIX THREADS

IEEE & POSIX:

- Institute of Electrical and Electronics Engineers ([IEEE](#))
- Portable Operating System Interface ([POSIX](#))

IEEE POSIX THREADS

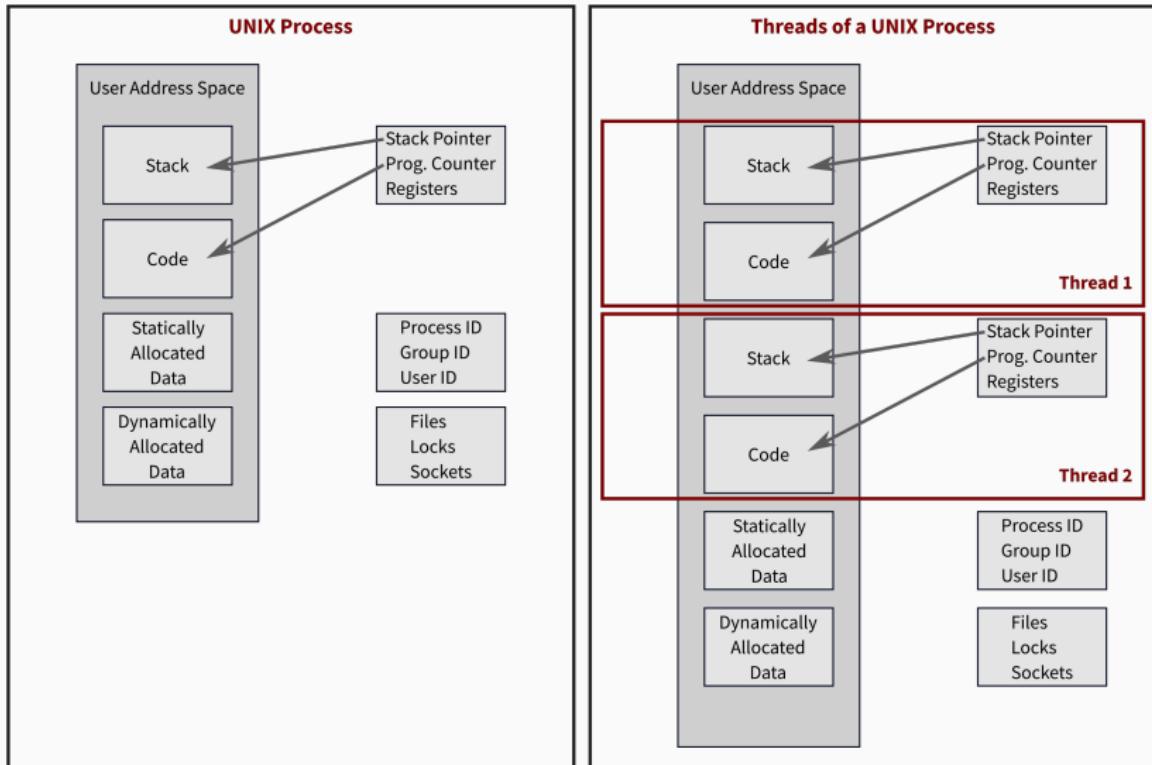
IEEE & POSIX:

- 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

PROCESSES & THREADS



PThreads API

~100 funções prefixadas por `pthread_`:

- Gerenciamento
- Mutexes
- Variáveis condicionais
- Sincronização

PThreads API

| 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://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);
};
```

POSIX THREADS: MAIS EXEMPLOS

Exemplos em <https://github.com/phrb/aula-pthreads-openmp>:

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

INTRODUÇÃO A PTHREADS & OPENMP

Pedro Bruel

phrb@ime.usp.br

17 de abril de 2017



Instituto de Matemática e Estatística
Universidade de São Paulo

ROTEIRO

1. Motivação
2. IEEE POSIX Threads
3. OpenMP

SLIDES



Os slides e todo o código fonte estão no [GitHub](#):

- github.com/phrb/aula-pthreads-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

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

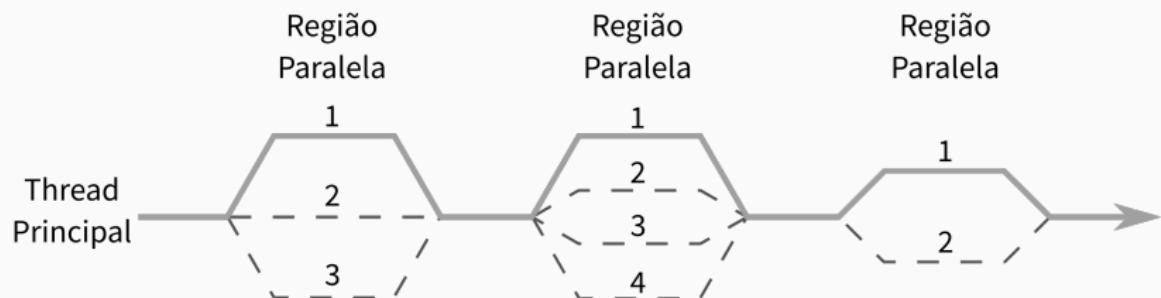
Objetivos:

- **Padronizar**
- **Simplificar**
- **Facilitar o uso**
- **Permitir portabilidade**

OPENMP: MODELO DE PROGRAMAÇÃO

- Threads dinâmicas
- Paralelismo explícito e aninhável
- Diretivas de compilador
- Modelo Fork-Join
- I/O e memória?

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

Usadas 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 em <https://github.com/phrb/aula-pthreads-openmp>:

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

OPENMP 4.5

Versão [4.5](#) da API OpenMP:

- Lançada em [Novembro de 2015](#)
- OpenMP [Architecture Review Board](#) (ARB)
- Requer [GCC 6](#):
 - Xeon Phi
 - Algumas GPGPUs AMD
 - GPGPUs Nvidia no futuro
- [Especificação completa](#) da API no repositório

OPENMP 4.5: EXEMPLO COM taskloop

Dividir e sincronizar 1024 iterações de um laço entre 32 *threads*, usando OpenMP < 4.5:

```
#pragma omp taskgroup
{
    for(int tmp = 0; tmp < 32; tmp++){
        #pragma omp task
        for(long l = tmp * 32; l < tmp * 32 + 32; l++){
            do_something (l);
        };
    };
}
```

OPENMP 4.5: EXEMPLO COM `taskloop`

Dividir e sincronizar 1024 iterações de um laço entre 32 *threads*, usando OpenMP ≥ 4.5 :

```
#pragma omp taskloop num_tasks(32)
for (long l = 0; l < 1024; l++){
    do_something(l);
};
```

OPENMP 4.5: EXEMPLO COM `taskloop`

Dividir e sincronizar 1024 iterações de um laço entre 32 *threads*, usando OpenMP \geq 4.5:

```
#pragma omp taskloop num_tasks(32)
for (long l = 0; l < 1024; l++){
    do_something(l);
};
```

Para saber mais, no [blog da RedHat](#):

<https://developers.redhat.com/blog/2016/03/22/what-is-new-in-openmp-4-5-3/>

INTRODUÇÃO A PTHREADS & OPENMP

Pedro Bruel

phrb@ime.usp.br

17 de abril de 2017



Instituto de Matemática e Estatística
Universidade de São Paulo