

Trabalho 1.1 - FORK

Dupla: Rita de Cássia Lino Lopes - T03

Társila Samille Santos da Silveira - T01

Passos para compilar:

1. Abrir terminal
2. Digitar o seguinte comando:
`g++ integral.cpp -o integral -pthread`

Passos para executar:

1. Após compilar o programa
2. No terminal, digitar o seguinte comando:
`./integral`
3. Digitar valores para t (número de thread) e n (número de trapézios).

Dados do Sistema Operacional:

Distributor ID:Ubuntu
Description: Ubuntu 20.04.2 LTS
Release: 20.04
Codename: focal

Versão do GCC:

9.3.0

Abaixo há uma cópia do código completo:

```
#include <iostream>
#include <cstdlib>
#include <pthread.h>
#include <math.h>

using namespace std;

#define PI 3.14159265

double a = 0, b = 10; // f1
//double a=0, b=2*PI; //f2

double h, sum = 0, integral;
int n, t, trapezio_por_thread;
double * x, * y;
```

```

struct thread_data {
    int thread_id;
};

double f1(double x) {
    return 5;
}
double f2(double x) {
    return sin(2.0 * x) + cos(5.0 * x);
}

void * fun(void * threadarg) {
    struct thread_data * my_data;
    my_data = (struct thread_data * ) threadarg;
    int inicio = my_data -> thread_id * trapezio_por_thread;
    int fim = (inicio + trapezio_por_thread) >= t ? t : (inicio + trapezio_por_thread);
    for (int i = inicio; i < fim; i++) {
        x[i] = a + i * h;
        y[i] = f1(x[i]);
        if (i!=0 && i!= t) sum += h * y[i];
    }
    pthread_exit(NULL);
}

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

    cout << "Digite o numero de threads:";
    cin >> n;
    cout << "Digite o numero de trapezios:";
    cin >> t;

    int rc;
    pthread_t threads[n];
    struct thread_data td[n];
    h = (b - a) / t;
    trapezio_por_thread = ceil((double)t / n);
    x = new double[t];
    y = new double[t];

    for (int i = 0; i < n; i++) {
        td[i].thread_id = i;
        rc = pthread_create( & threads[i], NULL, fun, (void * ) & td[i]);
        if (rc) {
            cout << "Error," << rc << endl;
            exit(-1);
        }
    }

    for (int i = 0; i < n; i++) {
        pthread_join(threads[i], NULL);
    }

    integral = h / 2.0 * (y[0] + y[t - 1]) + sum;
    cout.precision(2);
    cout<<scientific;
    cout << "\nA integral é " << integral << endl;

    pthread_exit(NULL);
}

```

As figuras 1 e 2 mostram, respectivamente, o resultado da execução dos testes. O teste com número de threads igual a 6 e de trapézios a 120 e o segundo teste com 7 threads e 200 trapézios.

```
37 tarsila@tarsila-Aspire-A315-42G:~/Área de Trabalho/so$ g++ integral.cpp -o integ
38 ral -pthread
39 tarsila@tarsila-Aspire-A315-42G:~/Área de Trabalho/so$ ./integral
40 Digite o numero de threads:6
41 Digite o numero de trapezios:120
42
43 A integral é 5.00e+01
44 tarsila@tarsila-Aspire-A315-42G:~/Área de Trabalho/so$ ./integral
45 Digite o numero de threads:7
46 Digite o numero de trapezios:200
47
48 A integral é 5.00e+01
49
50 tarsila@tarsila-Aspire-A315-42G:~/Área de Trabalho/so$
51
52
53
```

Figura 1: Resultado dos testes na função 1

```
37
38 tarsila@tarsila-Aspire-A315-42G:~/Área de Trabalho/so$ g++ integral.cpp -o integ
39 ral -pthread
40 tarsila@tarsila-Aspire-A315-42G:~/Área de Trabalho/so$ ./integral
41 Digite o numero de threads:6
42 Digite o numero de trapezios:120
43
44 A integral é -3.63e-03
45 tarsila@tarsila-Aspire-A315-42G:~/Área de Trabalho/so$ ./integral
46 Digite o numero de threads:7
47 Digite o numero de trapezios:200
48
49 A integral é -9.07e-02
50
51 tarsila@tarsila-Aspire-A315-42G:~/Área de Trabalho/so$
52
53 x = new double[1];
```

Figura 2: Resultado dos testes na função 2