



DCC - DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

RELATÓRIO TÉCNICO FINAL

PROGRAMAÇÃO MULTITHREAD

UFBA

Alunos: Itamar Joire, Luiz Gustavo e Osmar Pinto

Curso: Sistemas de informação

Professor: Alirio Sá

Salvador

Maio / 2022



DCC - DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

RELATÓRIO TÉCNICO FINAL
UFBA

Trabalho de fortalecimento de aspectos teóricos e práticos relacionados a programação envolvendo múltiplos threads por processo.

Salvador
Maio / 2022



DCC - DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

SUMÁRIO

1.	INTRODUÇÃO	4
2.	CONJECTURA DE COLLATZ	5
3.	FIBONACCI	6
4.	VALIDADOR SUDOKU	7
5.	ORDENAÇÃO (CLASSIFICAÇÃO)	8
6.	REPOSITÓRIO COM AS IMPLEMENTAÇÕES	9
7.	CONCLUSÃO	10
8.	REFERÊNCIAS BIBLIOGRÁFICAS	11



DCC - DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

1. INTRODUÇÃO

Os problemas e suas implementações que serão apresentadas a seguir trazem um entendimento inicial da abordagem de programação com o uso de multithreads. Para cada solução foi utilizado as primitivas básicas para manipulação de threads do Padrão Posix Threads (PThreads).



DCC - DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

2. CONJECTURA DE COLLATZ

Lothar Collatz conjecturou que, se você começar com um número inteiro positivo e executar esse processo por tempo suficiente, todos os valores iniciais levarão a **1**. E quando você chegar a 1, as regras da conjectura de Collatz farão você entrar em um ciclo repetitivo para sempre: **1, 4, 2, 1, 4, 2, 1...**

Regras da sequência – Dado um número inteiro positivo inicial:

- Se o número é par, dividir por **2**
- Se é ímpar, multiplicar por **3** e somar **1**
- Repita os passos acima com o resultado até atingir **1**

Na sua implementação a função principal cria uma única thread que faz o uso da fórmula de Collatz e mostra em tela o resultado como mostra a execução da imagem abaixo:

```
Atividades Terminal 17 de mai 17:38
joire@joire-VirtualBox: ~/Área de Trabalho/SO
joire@joire-VirtualBox:~/Área de Trabalho/SO$ gcc collatz.c -o collatz -lpthread
joire@joire-VirtualBox:~/Área de Trabalho/SO$ ./collatz
Qual o valor: 35
Main: criando uma thread para o cálculo de Collatz
35 106 53 160 80 40 20 10 5 16 8 4 2 1
Aguardando a finalização da thread...
joire@joire-VirtualBox:~/Área de Trabalho/SO$
```



DCC - DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

3. FIBONACCI

A famosa sequência de Fibonacci é uma sequência matemática composta por números inteiros. Normalmente, começa por 0 e 1 e cada termo subsequente é formado pela soma dos dois anteriores. Essa sequência é uma sucessão infinita de números que seguem o mesmo padrão.

Os números de Fibonacci compõem a seguinte sequência:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2594...

Como se vê, a composição é formada por números que são o resultado da soma dos dois anteriores:

- $0 + 1 = 1$
- $1 + 1 = 2$
- $2 + 1 = 3$
- $3 + 2 = 5$
- $5 + 3 = 8$

O programa implementado solicita quantos números de fibonacci deverá ser mostrado na sua execução. Assim que possuir o dado de entrada é criado uma thread que fará a sequência e retornará os valores. A thread-pai espera o retorno e mostra em tela. Como o exemplo da imagem abaixo:

```
Atividades Terminal 17 de mai 21:45
joire@joire-VirtualBox: ~/Área de Trabalho/SO
joire@joire-VirtualBox:~/Área de Trabalho/SO$ gcc fibonacci.c -o fibonacci -lpt
hread
joire@joire-VirtualBox:~/Área de Trabalho/SO$ ./fibonacci
Quantos números de Fibonacci deve aparecer? 10
Main: criando uma thread para gerar o Fibonacci
2 3 5 8 13 21 34 55 89 144
joire@joire-VirtualBox:~/Área de Trabalho/SO$
```

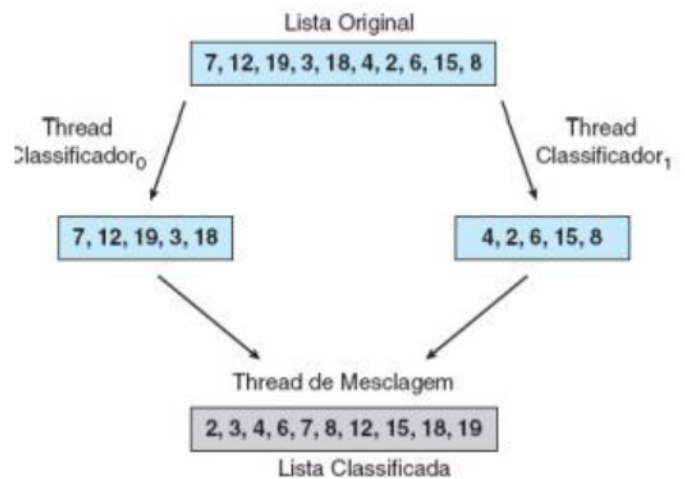



DCC - DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

5. ORDENAÇÃO (CLASSIFICAÇÃO)

Na computação existe uma série de algoritmos que utilizam diferentes técnicas de ordenação para organizar um conjunto de dados, eles são conhecidos como *Métodos de Ordenação* ou *Algoritmos de Ordenação*. Nesta implementação foi utilizado o **Selection Sort** que consiste em selecionar o menor item e colocar na primeira posição.

Para esta implementação foram criados **três threads** no qual a primeira tinha a função de ordenar uma metade do vetor e a segunda a outra metade. E a terceira thread une as duas threads anteriores em um novo array e ordena-o. Como mostra a figura ao lado:



A thread-pai mostra em tela o novo vetor ordenado como mostra a figura ao lado:

```
joire@joire-VirtualBox: ~/Área de Trabalho/threads-sistema...
joire@joire-VirtualBox:~/Área de Trabalho/threads-sistemas-operacionais/ordenac
ao-multithread$ gcc ordenacao.c -o ordenacao -lpthread
joire@joire-VirtualBox:~/Área de Trabalho/threads-sistemas-operacionais/ordenac
ao-multithread$ ./ordenacao
Main: Criando THREAD 1 de ordenamento...
Main: Criando THREAD 2 de ordenamento...

Aguardando as threads 1 e 2 concluírem.

Thread ordenador 2: 2 4 6 8 15
Thread ordenador 1: 3 7 12 18 19

Main: Criando THREAD 3 de ordenamento global...
Thread ordenador 3: 2 3 4 6 7 8 12 15 18 19
joire@joire-VirtualBox:~/Área de Trabalho/threads-sistemas-operacionais/ordenac
ao-multithread$
```




DCC - DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

6. REPOSITÓRIO COM AS IMPLEMENTAÇÕES

- [Repositório](#)



DCC - DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

7. CONCLUSÃO

O uso da programação multithread trouxe otimização para a resolução dos problemas, fazendo com que cada thread aja de forma independente e possua uma única função. Não existe a ideia de um programa, mas de threads (linhas). O processo, neste ambiente, tem pelo menos um thread de execução, podendo compartilhar o seu espaço de endereçamento com inúmeros threads, que podem ser executados de forma concorrente e/ou simultânea, no caso de múltiplos processadores. Se faz necessário o uso quando se precisa de desempenho, eficiência e economia de recursos do sistema como um todo.



DCC - DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

8. REFERÊNCIAS BIBLIOGRÁFICAS

- FIBOCACCI - [Link](#). Redação Brasil Paralelo.
- CONJECTURA DE COLLATZ - [Link](#). Brasil acadêmico.
- SUDOKU - [Link](#). Tagesspiegel Sudoku.
- ORDENAÇÃO (CLASSIFICAÇÃO) - [Link](#). Treina Web.