 INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA TRIÂNGULO MINEIRO	Curso: Bacharelado em Ciência da Computação		
	Unidade Curricular: Sistemas Operacionais	Ano/Período:	2022 / 7
	Tipo de Atividade: Trabalho #1	Aplicação:	25/05/2022
	Professor: Getúlio de Moraes Pereira	Entrega:	01/05/2022

1. Escreva um programa com múltiplos *threads* que exiba números primos. Esse programa deve funcionar da seguinte forma:

- O usuário executará o programa e dará entrada em um número na linha de comando.
- Em seguida, o programa criará um *thread* separado que exibirá todos os números primos **menores ou iguais** ao número informado pelo usuário.

Ex.:

> primos 10

> 1 2 3 5 7

2. Escreva um programa de classificação com múltiplos *threads* que funcione da seguinte forma:

- Uma lista de inteiros *e?* dividida em duas listas menores de **tamanho igual**.
- Dois *threads* separados (que chamaremos de *threads* classificadores) classificam cada sublista usando um algoritmo de classificação à sua escolha.
- As duas sublistas são, então, mescladas por um terceiro *thread* – um *thread* de mesclagem – que as combina em uma única lista classificada.
- Uma vez que dados globais são compartilhados por todos os *threads*, talvez a maneira mais fácil de configurar os dados seja criar um array global. Cada *thread* classificador trabalhará em uma metade desse *array*.
- Um segundo *array* global, com o mesmo tamanho do *array* de inteiros desclassificados, também será definido.
- O *thread* de mesclagem combinará as duas sublistas nesse segundo *array*.

Graficamente, esse programa tem a estrutura mostrada na figura 1.

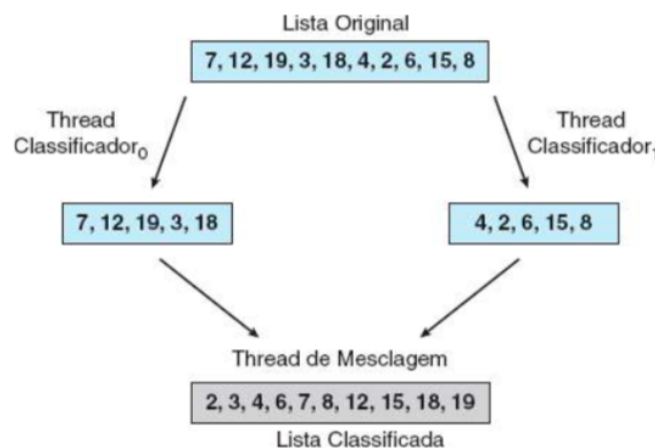


Figura 1: Classificação com múltiplas *threads*

Esse projeto de programação demandará a passagem de parâmetros para cada um dos *threads* classificadores. Especificamente, será necessário identificar o índice inicial a partir do qual cada *thread* deve começar a classificar. O *thread*-pai criará os *threads* classificadoras, assim como a *thread* de mesclagem. Possivelmente, a forma mais fácil de passar os parâmetros para as *threads* classificadoras seja por meio de uma estrutura de dados com o uso de *struct*. Por exemplo, uma estrutura para passar o índice de início e de fim no *array* dos dados originais para uma *thread* classificadora teria essa aparência:

```
/* estrutura para passagem de dados aos threads classificadoras */
typedef struct {
    int inicio;
    int fim;
} parameters;
```

A criação das *threads* pode seguir a estratégia abaixo:

```
parameters *data = (parameters *) malloc(sizeof (parameters));
data->inicio = 0;
data->fim = 4;
/* Cria agora o thread passando a ele data como parametro */
```

O *thread*-pai exibirá o *array* classificado, uma vez que todos os *threads* classificadores e de mesclagem tenham terminado.