

COMO COMPILAR

Compilar : gcc -pthread -o main main.c

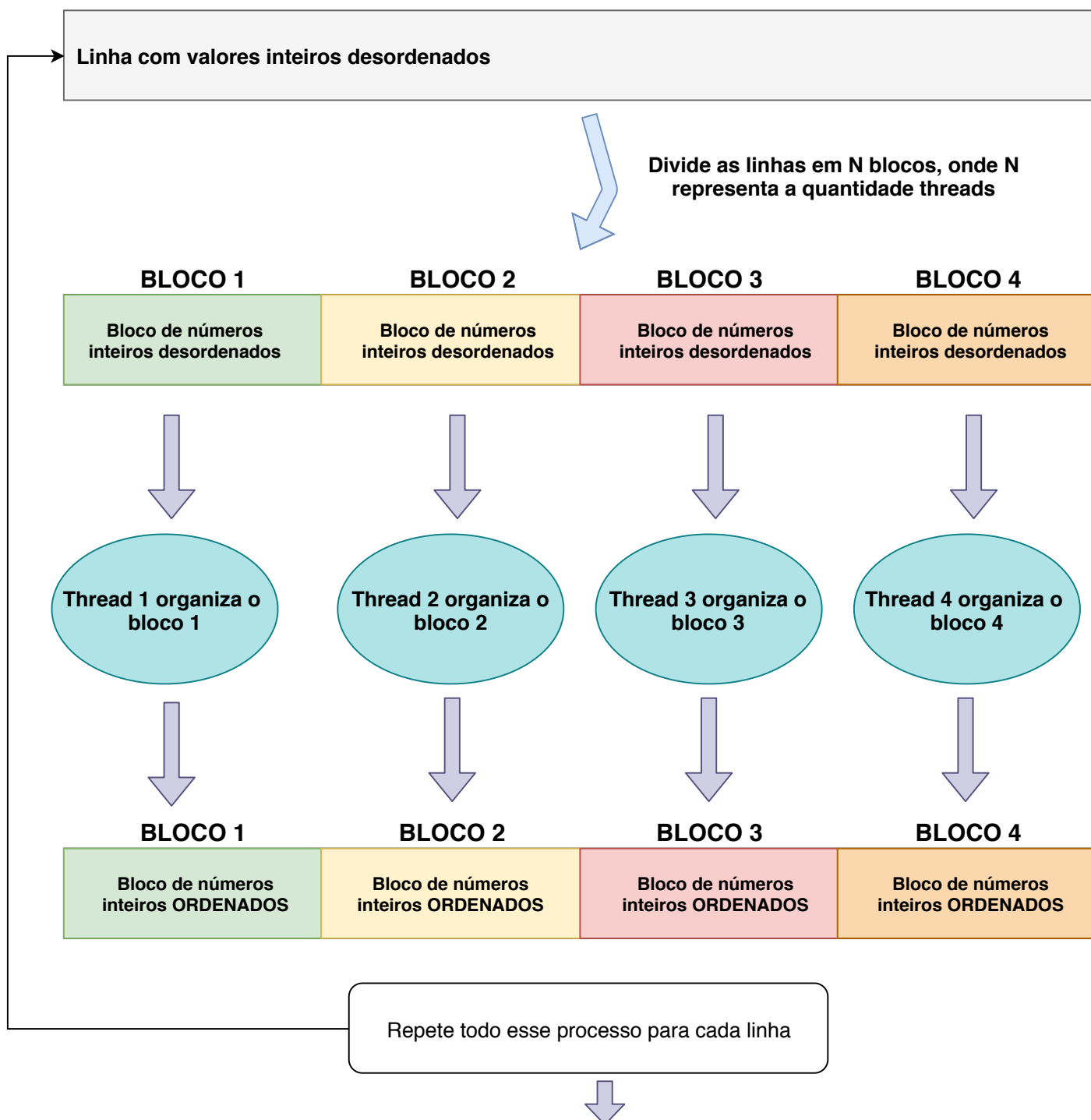
Executar: ./main <numero de threads> <arquivos de entrada> -o <arquivo de saída>

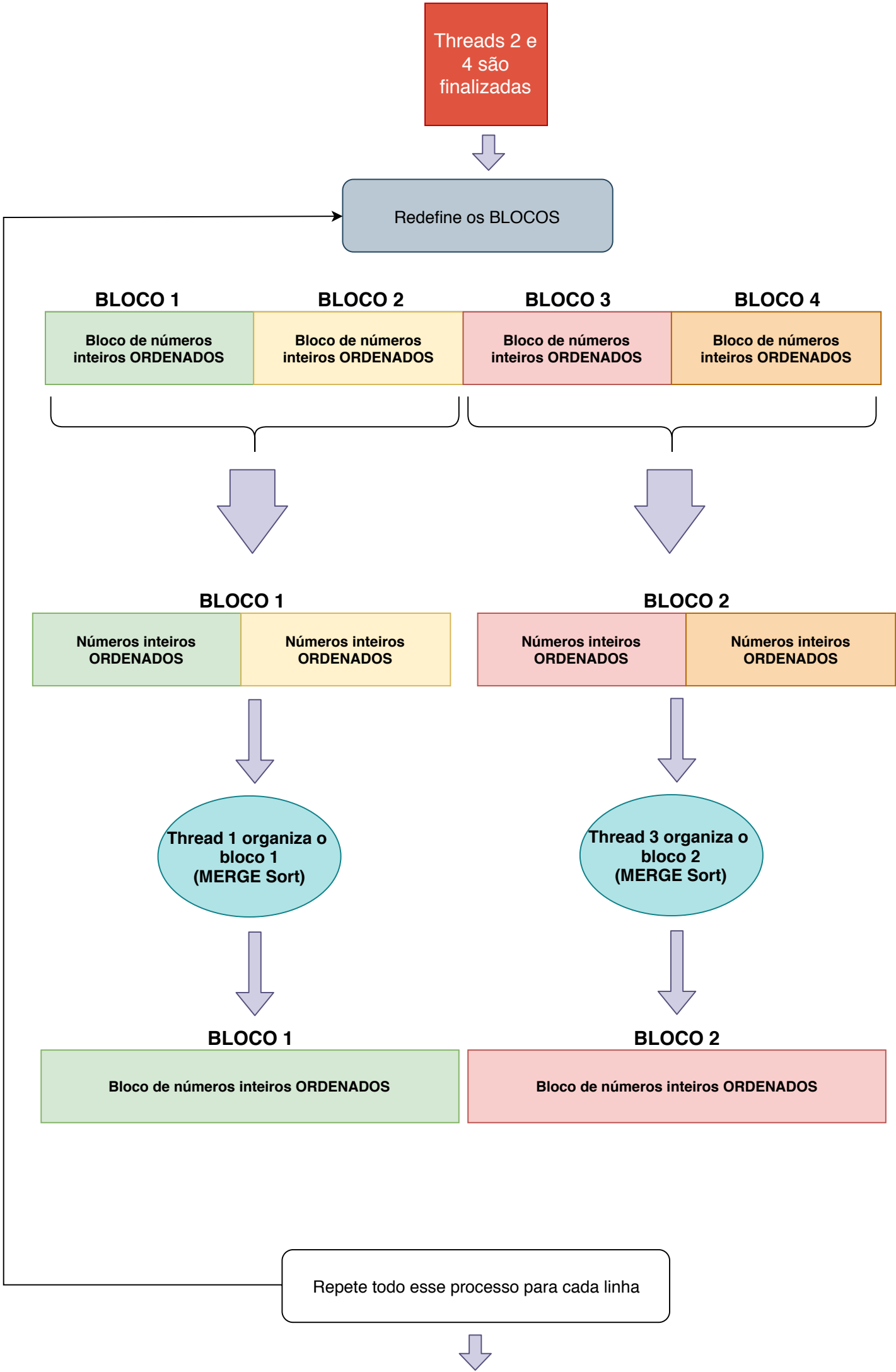
Exemplo de execução: ./main 8 arq1.dat arq2.dat arq3.dat -o saida.dat

GITHUB : <https://github.com/ticollapse/threads-projeto2>

DIAGRAMA COM SOLUÇÃO DO PROBLEMA

Carrega uma matriz com X linhas, onde X representa a quantidade de arquivos de entrada





Thread 3 é finalizada



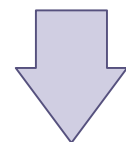
Redefine os BLOCOS

BLOCO 1

Bloco de números inteiros ORDENADOS

BLOCO 2

Bloco de números inteiros ORDENADOS



BLOCO 1

Bloco de números inteiros ORDENADOS

Bloco de números inteiros ORDENADOS



Thread 1 organiza o
bloco 1
(MERGE Sort)



LINHA de números inteiros ORDENADOS

Repete todo esse processo para cada linha



Thread 1 é finalizada



Fim

GRÁFICOS COM OS TEMPOS DE EXECUÇÃO

O teste foi realizado 5 vezes com 37 arquivos de entrada, que continham 5.000 números inteiros cada.

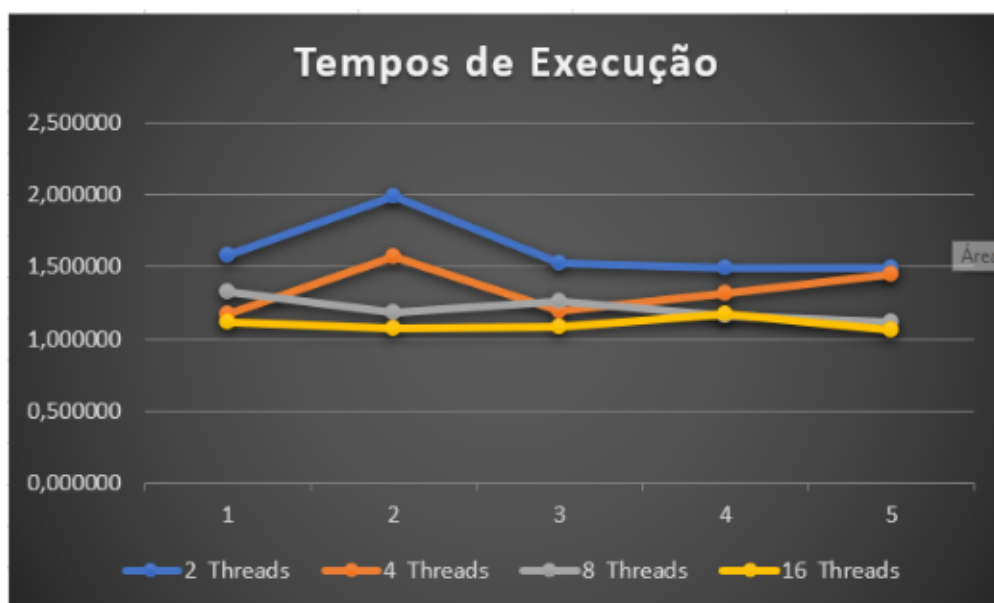


Gráfico 2D

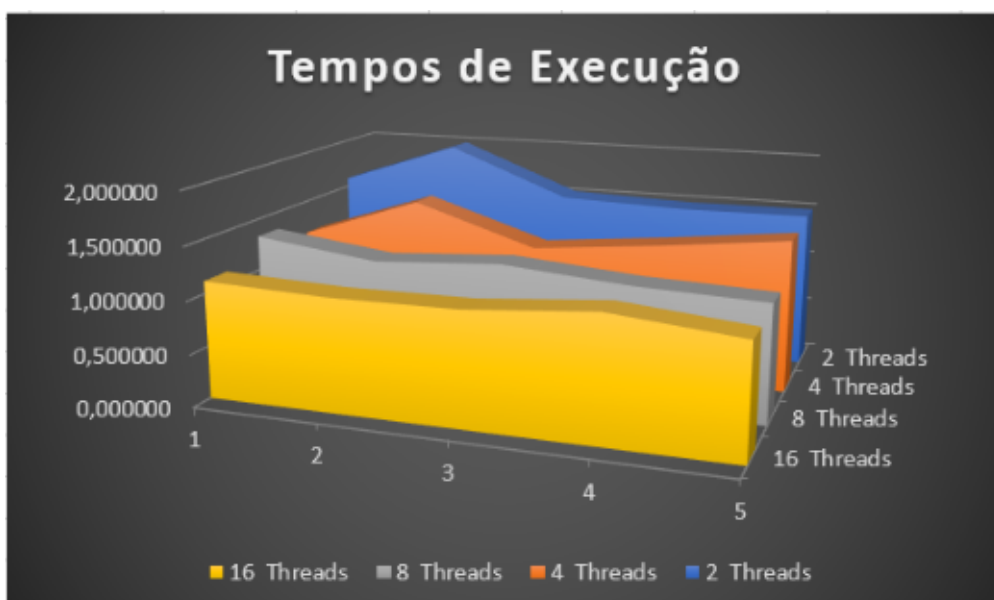


Gráfico 3D

CONCLUSÃO

Conforme o numero de threads aumenta o tempo de execução do programa diminui.

Porém o tempo não diminui de forma proporcional ao aumento de número de threads, provavelmente devido ao fato de que o computador em que foi realizado esses testes possuía apenas 6 núcleos e foi utilizado uma maquina virtual para emular o sistema operacional Ubuntu.

Portanto para que o programa consiga melhorar o desempenho conforme o numero de threads aumenta é necessário que o computador tenha núcleos de processamento suficiente.