# Otimização do Programa da Aula 10

**Desenvolvida por**: Raul A. Gonzalez Augusto, RA.: 211023698

Objetivo: O exercício de hoje consiste em transformar uma imagem para preto e branco, processando cada pixel através de ponteiros e comparando os tempos com o processamento feitos usando números com ponto flutuante e com inteiros considerados como números reais de ponto fixo. Como os inteiros são bem mais simples para calcular, costumam ser mais rápidos que com reais.

# Computador Usado:

Avell High Performance A52 LIV

* Processador: Intel® Core™ i5-10300H CPU @ 2.50GHz
  + Cores: 4
  + Threads: 8
  + Frequência Máxima: 4.50 GHz
  + Cache: 8 MB Intel® Smart Cache
  + Bus Speed: 8 GT/s
* Placa de Vídeo: NVIDIA GeForce GTX 1650 Ti e Intel® UHD Graphics
* RAM: 16 GB DDR4-2666 Dual Channel
* Armazenamento: 500 GB nvme m.2 SSD 2GB/s E 1 Tera HDD 5400 rpm
* Sistema Operacional: Windows 10 pro 64bits

# Programas

Compilado no gcc em uma máquina virtual Linux.

# Executando o programa sem otimização

Tempos:

0.206856

0.209542

0.204436

0.208035

0.206084

Media: 0.2069906

# Programa com ponteiros

Tempos:

0.210297

0.207361

0.210662

0.209007

0.206993

Media: 0.208864

como é possível transformar cada pixel da imagem utilizando apenas um ponteiro para a imagem de entrada e outro para a imagem de saída?

A matriz que guarda a imagem é armazenada em um espaço da memória, que começa em um endereço e os endereços seguintes são dos elementos da matriz ate chegar ao ultimo elemento, podemos apontar um ponteiro para o primeiro elemento da matriz e ir incrementando ele, assim ele vai percorrer cada elemento da matriz de forma sequencial já que os elementos estão guardados em endereços de forma sequencial.

# Programa usando inteiro invés de ponto flutuante

Tempos:

0.135721

0.134081

0.135492

0.135175

0.135268

Media: 0.1351474

Aumento de performance em relação ao original: 1.5x de speedup

# Programa usando linguagem de maquina

Tempos:

0.128469

0.132815

0.132583

0.131098

0.130259

Media: 0.1310448

como funciona o modo de endereçamento utilizado nas linhas e para que estas linhas servem no programa?

" MOV [RDI+RDX],AL \n"

" MOV [RDI+RDX+1],AL \n"

" MOV [RDI+RDX+2],AL \n"

Esta endereçando o início até um deslocamento, RDI início, + RDX o deslocamento, as outras duas linhas tem um deslocamento + 1 e + 2 para ir para os canais azul e verde.