EA876 - Trabalho 2

O problema com o qual vamos lidar nesse trabalho é a paralelização de filtros de imagem.

Uma imagem é representada, na máquina, por três matrizes. Cada uma delas se refere a um canal (R, G, B) da imagem. Então, r[i][j], g[i][j] e b[i][j] se referem aos valores de R, G e B do pixel (i,j). A imagem inteira tem tamanho I por J.

Num filtro blur de tamanho N, novas matrizes r', g' e b' são geradas. Para cada uma delas, o pixel (i,j) recebe o valor da média de todos os pixels da matriz original que estão dentro do quadrado que vai de (i-N, j-N) a (i+N, j+N).

r' (i,j) = media de r(i-N até i+N, j-N até j+N)

O objetivo técnico do trabalho é fazer um programa de computador que permita identificar se a aplicação de um filtro tipo "blur" em uma imagem é mais rápida se ocorrer em uma única linha de execução, em múltiplas threads ou em múltiplos processos.

Para tal, o grupo deverá implementar as três propostas e então avaliar objetivamente seu tempo de execução em imagens pequenas e em imagens grandes.

Restrições

1. Este trabalho deverá ser feito em C. Há, no repositório da disciplina, código para abrir, manipular e salvar imagens.
2. O grupo, porém, deve se sentir livre para usar outras bibliotecas ou outras soluções para abrir imagens.
3. Não é permitido usar bibliotecas que implementem filtros de imagens (essa parte deverá ser codificada pelo próprio grupo).

O que entregar

1. O código-fonte deverá ser entregue comentado de forma que seja possível entender o raciocínio do grupo através dos comentários.
2. Um arquivo Makefile e instruções de uso para permitir testar o programa enviado.
3. O grupo deverá entregar uma figura em formato PDF mostrando a média e desvio padrão dos tempos de execução de cada uma das variações do programa. Recomendamos executar ao menos 100 vezes cada variação, buscando encontrar dados estatisticamente significativos.
4. O código e a figura deverão ser entregues num arquivo .zip através do sistema do Classroom. Por favor, deixe a figura mostrando o tempo de execução num diretório chamado **doc,** as imagens usadas para teste no diretório **data**, os códigos fontes num diretório chamado **src** e o Makefile no diretório raiz.

Notas

- A nota será dada com base no código, que inicia em 10 e considera os seguintes descontos:

- -10 se o programa não compila ou não executa (ele será testado em máquinas Linux!)
- -4 para cada versão não implementada (devem ser entregues as versões multi-thread, multi-processo e em linha de execução única).
- -2 se não há script de teste
- -2 se não há instruções para execução
- -2 se o programa tem bugs que poderiam ser resolvidos facilmente (exemplo: overflow nos pixels levando a aberrações cromáticas)
- -2 se o programa não tem comentários, de forma a torná-lo ilegível.
- +2 se o comando *make test* compila o programa, executa automaticamente todos os testes e gera automaticamente a figura com os resultados (sugestões: Python, GNUPlot).

Variações

É possível que o grupo prefira fazer outro filtro que não seja o "blur", ou queira implementar uma variação do "blur". Alternativamente, o grupo pode optar por implementar outros filtros, tais como:

- Detector de bordas (filtro de Sobel)
- "Blur" Gaussiano
- Erosão
- Dilatação
- Mediana, máximo ou mínimo em janela deslizante

Todos esses filtros têm a característica de depender de uma área em torno de cada pixel processado. Em tese, qualquer filtro com essa característica é possível para este trabalho, mas pode ser uma boa idéia conversar com o Tiago e com o Rafael se o grupo tiver uma outra idéia de filtro.