

**Instituto Federal de Educação, Ciência e Tecnologia do Sul de Minas –
Campus Passos
Bacharelado em Ciência da Computação
Programação Paralela
Profª. Francielli Barbara Pinto**

Trabalho Prático 1 – programação paralela em Java

I – Observações:

1) Objetivos: o objetivo deste trabalho é colocar em prática o que foi visto acerca da programação multitarefa e interprocesso em Java.

2) O trabalho poderá ser feito em duplas ou individualmente. É permitido discutir os problemas e estratégias de solução com outros colegas/duplas, mas quando se tratar de implementar computacionalmente as soluções, isto deve ser feito separadamente.

3) Forma de entrega: O trabalho deve ser entregue em formato digital por meio do *Google Sala de Aula*. Utilizar a atividade *"Programação paralela em Java"*. Anexe **um único arquivo .zip** contendo todos os arquivos da pasta *src* do seu projeto Java (ou a pasta) e um arquivo .txt contendo o nome completo do aluno ou dos integrantes da dupla. Para trabalhos realizados em dupla, basta que somente um integrante faça o envio.

4) Trabalhos copiados receberão nota zero para todas as cópias. Trabalhos com erros de compilação não serão avaliados e receberão nota zero. Os programas devem ser desenvolvidos seguindo as boas práticas de programação e em linguagem Java.

5) Data limite para entrega: 20/12 às 23h55.

6) Valor: 2,5 pts

Tarefa:

Alguns programas de processamento de imagens exigem um algoritmo que consiste em transformar uma imagem colorida em uma imagem em tons de cinza. Em uma imagem, cada pixel possui uma cor que pode ser representada pelo padrão RGB, sigla que representa as cores primárias: vermelho (Red), verde (Green) e azul (Blue). Qualquer cor pode ser formada pela

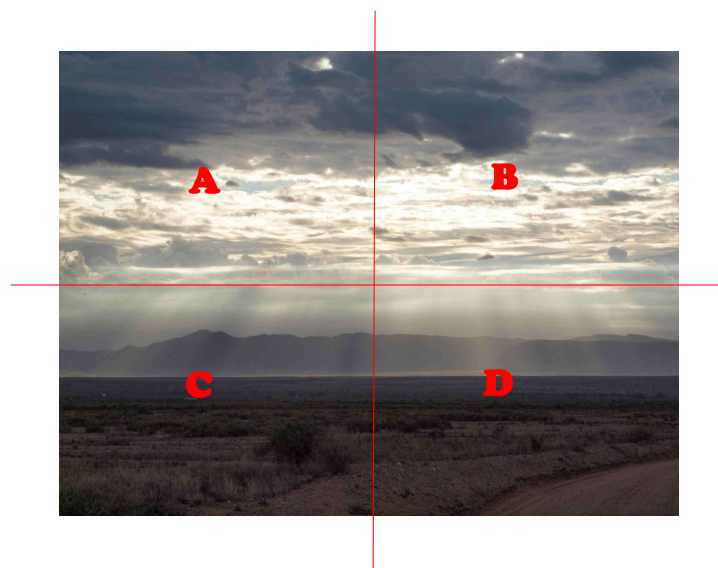
combinação destas três cores. A cor amarela, por exemplo, pode ser formada pela combinação das cores verde e vermelho. Neste padrão, cada cor primária possui 256 tons (intervalo entre 0 e 255, números inteiros), onde quanto mais alto o tom, mais “escuro” é o tom. Um das formas de representar a cor amarela, dessa forma, seria {123; 123; 0}, o que corresponde, respectivamente, aos tons de vermelho, verde e azul. A cor preta é a ausência de cor, sendo representada, portanto, no padrão RGB, por {0; 0; 0} e a cor branca é a combinação de todas as cores, sendo apresentada, no padrão RGB por {255; 255; 255}.

A conversão de uma imagem para escalas de cinza consiste em converter cada pixel da imagem para um tom de cinza. Uma cor é um tom de cinza quando os tons de vermelho, verde e azul são iguais. {111; 111; 111} é, portanto, um tom de cinza. Uma das formas de se fazer essa conversão é pelo método da média, que consiste em encontrar o tom de cinza através da equação $(R + G + B) / 3$, onde R, G e B seriam os tons de vermelho, verde e azul do pixel, respectivamente, obtendo-se, então, a média dos tons de vermelho, verde e azul.

Considerando, como exemplo, o pixel em tom de amarelo apresentado acima, converter esse pixel para um tom de cinza seria, primeiramente, obter a média dos tons RGB: $(123 + 123 + 0) / 3 = 83$. E depois atribuir aos tons RGB do pixel o valor resultante. O pixel em tom de cinza resultante, portanto, no padrão RGB, ficaria {82; 82; 82}.

Ainda, é comum que programas que realizem processamento de imagens dividam uma tarefa entre múltiplas *threads*. Tem-se, portanto, um programa com múltiplas partes (*threads*) que cooperam entre si para resolver um problema. Então, para resolver um problema desse tipo, por exemplo, um programa poderia criar 4 *threads* e dividir a tarefa de converter cada pixel da imagem para um tom de cinza entre tais *threads*, dividindo a imagem original em 4 quadrantes, em que cada *thread* ficaria responsável pela conversão dos pixels de determinado quadrante.

Considere como exemplo a imagem a seguir:



Para converter tal imagem em tons de cinza, um programa poderia, por exemplo, criar as *threads* 1, 2, 3 e 4 e a *Thread* 1 poderia se encarregar de converter os pixels do quadrante A da imagem para tons de cinza, enquanto que a *Thread* 2 ficaria responsável pela conversão dos pixels do quadrante B e assim por diante.

Considerando esse problema, para fins de praticar a programação multitarefa e interprocesso em Java, você deverá:

1. Criar um Cliente que se conecta a um Servidor e envia a esse Servidor uma imagem colorida para que a mesma seja convertida para tons de cinza – para isso, você deve pesquisar como enviar arquivos utilizando Sockets – e aguarda uma resposta do Servidor. O caminho da imagem pode ser informado manualmente pelo programador via código ou ainda pelo usuário ao executar o programa;
2. Criar um Servidor que aguarda a conexão com um Cliente e, tendo sido estabelecida uma conexão com um Cliente, aguarda o mesmo enviar-lhe uma imagem;
3. Ao receber a imagem, o Servidor deve proceder com a conversão da imagem para tons de cinza, conforme descrito acima – você deverá pesquisar como obter os valores RGB para cada pixel de uma imagem (Dica: para resolver parte do problema, você poderá usar as classes *java.awt.Color* e *java.awt.image.BufferedImage*). Considere, para isso, que o trabalho será dividido entre 4 (quatro) *threads* – o Servidor, portanto, iniciará 4 *threads* e cada uma procederá com a conversão dos pixels de um quadrante da imagem;
4. O Servidor deverá enviar a imagem resultante para o Cliente;
5. O Cliente, ao receber a imagem resultante, deverá salvá-la em uma pasta de nome “imagensResultantes” – isso é interessante para que a imagem original não seja sobrescrita –, finalizando, assim, sua execução.

Por último, considere que você não precisará se preocupar com controle de acesso/concorrência para a resolução desse problema e que o Servidor pode tratar apenas um único Cliente.

Qualquer dúvida, procurem a professora responsável.

Bom trabalho!