

Programação mista C++/Assembly

NOTA: Os exercícios seguintes propõem o desenvolvimento de sub-rotinas para incluir no programa VIEWER. É fundamental ler o documento “Programa de tratamento de imagem” antes de prosseguir.

1. Pretende-se efetuar alterações simples à cor de uma imagem. Para cada uma das alíneas seguintes, escreva uma sub-rotina que implemente a transformação descrita, e experimente o efeito numa imagem.
 - a) Eliminar uma componente RGB, isto é, atribuir-lhe o valor 0.
 - b) Trocar as componentes RGB: $R \leftarrow G$, $G \leftarrow B$ e $B \leftarrow R$.
 - c) Aumentar a intensidade da cor, somando uma constante às componentes de cor.
2. Identificar e alterar pixels numa imagem são operações frequentes.
 - a) Escreva uma sub-rotina que permita localizar um pixel (*coluna*, *linha*) e alterar a sua cor. Comece por calcular a posição desse pixel em memória, atendendo a que **largura** e **altura** definem a dimensão da imagem, alterando-o de seguida para preto.
 - b) Marcar a vermelho ($RGB = \{255, 0, 0\}$) o pixel mais intenso em cada linha. Considere que o pixel mais intenso é o que possui maior valor médio das componentes de cor.
3. As operações de reflexão geométrica são comuns no processamento de imagem.
 - a) Implemente uma sub-rotina que realize uma reflexão horizontal, isto é, que inverta cada linha da imagem por permuta dos respetivos pixels.
 - b) Implemente uma sub-rotina que realize uma reflexão vertical.
4. Escreva uma sub-rotina capaz de transformar uma imagem a cores numa imagem binária em que todos os pixels são brancos ou pretos. Esta operação pode decompor-se em duas etapas:
 - transformar a imagem original numa imagem com níveis de cinzento;
 - tomando como referência um dado nível de cinzento, passar os pixels com cinzento mais escuro que essa referência para preto e passar os pixels com cinzento mais claro para branco.
5. Escrever uma sub-rotina que substitui cada pixel pela média do próprio pixel e dos seus oito vizinhos (para cada componente R, G e B). Ignorar os pixels situados na periferia da imagem.

Para obter memória auxiliar, usar a seguinte sub-rotina:

`malloc PROTO C numBytes:DWORD`

Esta função retorna o endereço inicial de uma zona de memória com `numBytes` em EAX. Para libertar a memória obtida com `malloc`, passar o endereço inicial da zona à sub-rotina `free`:

`free PROTO C memptr:BYTE PTR`