

Exercícios



Bitmap

Leia um arquivo bitmap e imprima na tela as informações sobre ele.

O arquivo bitmap é um arquivo binário que segue o seguinte formato:

https://en.wikipedia.org/wiki/BMP_file_format

Bitmap

Cabeçalho do arquivo:

ID	Tamanho em bytes	Reservado	Reservado	Endereço dos Dados
(2 bytes)	(4 bytes)	(2 bytes)	(2 bytes)	(4 bytes)

ID geralmente é a string “BM”

Bitmap

O tipo char tem 1 byte, o tipo int geralmente tem 4 bytes.

```
typedef struct BitmapHeader{  
    unsigned char signature[2];  
    unsigned int size;  
    char reserved[4];  
    unsigned int offset;  
} BitmapHeader;
```

Bitmap

Informação do Bitmap:

Tamanho do cabeçalho	Largura	Altura	Planos	Bits por pixel	Compressão
(4 bytes)	(4 bytes)	(4 bytes)	(2 bytes)	(2 bytes)	(4 bytes)

Bitmap

Informação do Bitmap:

Tamanho da imagem	Resolução Horizontal	Resolução Vertical	Num. Cores	Num. Cores Importantes
(4 bytes)	(4 bytes)	(4 bytes)	(4 bytes)	(4 bytes)

Bitmap

Informação do Bitmap:

Mascara cor vermelha	Máscara cor verde	Máscara cor azul	Máscara transparênc ia	Tipo de esquema de cor
(4 bytes)	(4 bytes)	(4 bytes)	(4 bytes)	(4 bytes)

Bitmap

Informação do Bitmap:

PAD	Gama vermelho	Gama verde	Gama azul
(36 bytes)	(4 bytes)	(4 bytes)	(4 bytes)

Bitmap

```
typedef struct DIB{  
    unsigned int sizeheader;  
    unsigned int width;  
    unsigned int height;  
    unsigned short planes;  
    unsigned short bpp;  
    unsigned int compression;  
    unsigned int imgsize;  
    unsigned int hppm;  
    ...  
}
```

Bitmap

```
typedef struct DIB{  
...  
    unsigned int alphamask;  
    unsigned int cs_type;  
    char pad[36];  
    unsigned int gamma_red;  
    unsigned int gamma_green;  
    unsigned int gamma_blue;  
} DIB;
```

Packing: removendo os paddings

Quando precisamos ler arquivos binários em estruturas pré-definidas, temos que avisar o compilador para não realizar os **paddings**.

Para isso utilizamos a instrução:

```
#pragma pack(1)
```

Na linha anterior a definição do struct.

Bitmap

Vamos trabalhar com dois esquemas de cores:

32 bits - Blue Green Red Alpha (4 bytes)

24 bits - Blue Green Red (3 bytes)

Bitmap

```
typedef struct Color32 {  
    unsigned char blue;  
    unsigned char green;  
    unsigned char red;  
    unsigned char alpha;  
} Color32;
```

```
typedef struct Color {  
    unsigned char blue;  
    unsigned char green;  
    unsigned char red;  
} Color;
```

Bitmap

Faça o download de **bitmap.h** e **bitmapinfo.c**, verifique como o programa faz para:

- abrir um arquivo bitmap no modo binário,
- ler o **header** e o **dib** utilizando a função **fread**
- imprimir: **assinatura, largura e altura, bits por pixel, tamanho da imagem, número de cores.**

Arte ASCII

Para ler os pixels de um bitmap, após a leitura do **header** e do **dib**, basta criar uma array da estrutura **Color** ou **Color32**, dependendo do esquema de cores com tamanho **dib.width*dib.height**.

Em seguida, leia essa quantidade de dados utilizando **fread**.

Arte ASCII

O pixel da posição i, j ($0 \leq i < \text{dib.height}$, $0 \leq j < \text{dib.width}$) de data pode ser acessada por:

$*(\text{data} + i * \text{dib.width} + j)$

ASCII

Se utilizarmos os caracteres `.,:;ox%#@` (o primeiro é o espaço em branco), para representar as diferentes intensidades do pixel, do mais claro ao mais escuro, podemos substituir cada pixel por uma sequência de duas repetições de um desses caracteres:

`##`

Para representar um pixel quadrado

Arte ASCII

A intensidade de cada pixel pode ser calculada por:

$$\text{CARACTERES} - (\text{Blue} + \text{Green} + \text{Red}) * \text{CARACTERES} / (255 * 3)$$

com CARACTERES representando quantos caracteres utilizaremos para substituir os pixels.

Arte ASCII

Com isso, faça o download de **bitmapEx.c** e complete no local indicado para:

- acessar o pixel na posição i,j
- somar os componentes red, green, blue do pixel, armazenando em **intensity**
- fazer com que **intensity** seja: **$\text{SIZEMAP} - \text{intensity} * \text{SIZEMAP} / \text{MAX_SUM_COLORS}$**
- imprimir duas vezes seguida o caractere em **map[intensity]**

Atividade Extra

Estude o arquivo **LeituraEscritaEstrutura.c**