

TP 14 : Fichiers

Préparation

Dans ce TP, nous allons nous intéresser à l'utilisation des fichiers. Il est effectivement possible en C de lire et d'écrire dans un fichier. Il existe deux formats de fichier : ASCII et binaires.

- Les fichiers ASCII sont lisibles partir d'un éditeur de texte. Par exemple un fichier source en C est un fichier au format ASCII, il est possible de l'ouvrir avec un éditeur :

```
#pragma hdrstop
#include <condefs.h>
#include <stdio.h>
#include <conio.h>
#include <math.h>

//-----
#pragma argsused

struct Date
{
    int Jour;
    int Mois;
    int Annee;
};
```

- Les fichiers binaires ne peuvent pas être lu avec un éditeur de texte et doivent être ouverts avec le logiciel approprié. Voici à quoi ressemble une image BMP ouverte avec un éditeur de texte :

```
%R<DC4@÷FF;E[ B0TF(é'÷ò~Î=â>&E2C"ûBTEACRGçMAR#i6Eê°â+SUBc°ñETX2
7{:Yc·Wë·q<0*STX:GHδ^qøSTX,,ãæI0e0Ã W<z4yÎ;ðSTXSYWSDHÇù~DC4Ûz '8«
6aŠ'S·FS' NAR'Y' $zNfYi$%MDdtÇÃš8BED'iAICMk"Oz"úŮZ
E5STISDrBo`ENQESTX:ùpRACK!s%ieoDA%NAR#ENQ.û,,SUBlfKwÍVT,CSUSÊCAN
p*5"is%3M±"QYUDC2Á"/%YESCmÎ" 'mZçUS×çBsŠpc#·Î·eStæ2%*Yk!CAN
Uç2ÿ*0»Xød'·k;ë/ESCæ0F°(N"ghDE<_ÃÃ,YlAô$Š'=iBNOSTNARósoeIM"\\ ,nE
»STX>ú?UT,PyIm*Á×$ê*9·!ÚeACRô060í^3BEDY
!sc
CÍCOX @xFFMKSTI0cDnæIeUS?·6US?+·FS=æqNAR°Öi0Ã.YÀ,NARrSTX8s>=D1B
+J#KL
çfs0lÉÿ7*40m;FS_[!5
a~ð*bN«"º3Ã³,Et,,FB=ÃÇ³·(z¹bEi%³CSz<i XFO"i×*GGAk/ðI CANeðÿÚNoóBS-q
10TY0W«·YË·é0Š"·æB¹·é«·FSZ0041FS50M1W0e
```

Dans la suite du TP, nous allons nous intéresser au format BitMaP (BMP) qui est un format de fichiers qui permet de coder des images. Ce sont des fichiers binaires constitués de deux parties : la première partie est l'entête ou header (elle-même constituée de deux parties), elle contient des informations relatives à l'image (taille de l'image, nombre de couleurs etc...). La seconde partie contient l'image proprement dite.

Le tableau suivant montre la structure d'une image BMP

Position		Taille	Désignation
Hexa	Décimal		
Première partie de l'entête, informations relatives au fichier			
0x00	0	2 octets	Magic number, utilisé pour identifier un fichier de type Bitmap. Dans le cadre du TP, ces octets doivent contenir 'B' et 'M'.
0x02	2	4 octets	Taille du fichier en octets.
0x06	6	2 octets	Réservé (0).
0x08	8	2 octets	Réservé (0).
0x0A	10	4 octets	Offset, position dans le fichier où l'image commence.
Seconde partie de l'entête, informations relatives à l'image			
0x0E	14	4 octets	Taille de la seconde partie de l'entête (ici 40 octets).
0x12	18	4 octets	Largeur de l'image en pixels.
0x16	22	4 octets	Hauteur de l'image en pixels.
0x1A	26	2 octets	Nombre de plan dans l'image. Dans le cadre de ce TP, nous n'utiliserons qu'un seul plan (1).
0x1C	28	2 octets	Nombre de bits pour coder un pixel. Dans le cadre de ce TP, les pixels seront codés sur 24 bits (8 bits pour le rouge, 8 bits pour le vert et 8 bits pour le bleu)
0x1E	30	4 octets	Méthode de compression utilisée. Dans le cadre de ce TP, il n'y aura pas de compression (0).
0x22	34	4 octets	Taille de l'image proprement dite.
0x26	38	4 octets	Résolution horizontale de l'image (pixel par mètre).
0x2A	42	4 octets	Résolution verticale de l'image (pixel par mètre).
0x2E	46	4 octets	Nombre de couleur dans la palette. Dans le cadre de ce TP, nous n'utiliserons pas de palette (0).
0x32	50	4 octets	Nombre de couleurs importantes, généralement ignoré (0).
Image			
0x36	54		Début de l'image (le codage commence par le coin en bas à gauche de l'image)

Soit les deux images suivantes : BMP_2x2.bmp et IUT.bmp dont les propriétés sont données ci-dessous :

	BMP_2x2.bmp	IUT.bmp
Hauteur en pixel	2	377
Largeur en pixel	2	600
Nombre de couleurs	16 millions	16 millions
Compression	Aucune	Aucune
Résolution	2835 pixels par mètre	2835 pixels par mètre

D'après les informations ci-dessus, compléter les entête sur la page suivante.

Position	BMP_2x2.bmp	IUT.bmp
0x00	BM	
0x02		
0x06	Réservé.	Réservé.
0x08	Réservé.	Réservé.
0x0A		
0x0E		
0x12		
0x16		
0x1A		
0x1C		
0x1E		
0x22		
0x26		
0x2A		
0x2E		
0x32	0	0

Travail pratique

Exercice 1 :

a. Saisir, compléter et commenter le programme suivant :

```
#define FILE_NAME "IUT.bmp" // Fichier à analyser

// Programme principal

int main()
{
    FILE *Img; // Flux de données vers le fichier
    char MagicNumber[2]; // Tableau pour le nombre magique
    int FileSize; // Taille du fichier
    int Offset; //
    short int NbBits; //
    int Compression; //
    int ImgSize; //
    int NbColors; //
    int HeaderSize; //
    int ImgDim[2]; //
    int ImgResolution[2]; //
```

```

// Ouverture du fichier

    Img=fopen (FILE_NAME,"rb"); //
    if (Img==NULL) //
    {
        printf ("Erreur lors de l'ouverture !");
        return -1;
    }

// Lecture du nombre magique

    fread (MagicNumber,1,2,Img);
    printf ("Magic number : %c%c\n",MagicNumber[0],MagicNumber[1]);

// Lecture de la taille du fichier

    fread (&FileSize,4,1,Img);
    printf ("Taille du fichier : %d octets\n",FileSize);

    // ... A completer ...

// Fermeture du fichier

    fclose (Img);
    getch();
    return 0;
}

```

b. Expliquer ce que signifie l'option "rb" lors de l'ouverture du fichier :

```

Img=fopen (FILE_NAME,"rb");

```

c. Expliquer chacun des paramètres lors de la lecture des données :

```

fread (MagicNumber,1,2,Img); // Lecture du nombre magique
fread (&FileSize,4,1,Img); // Lecture de la taille du fichier

```

d. Relever les tailles des fichiers pour chacune des deux images. Pour des raisons pratiques, chaque ligne de l'image doit être mémorisé sur un multiple de 4 octets. Justifier pour chacune des images les tailles relevées.

e. Compléter le programme afin d'afficher et de relever sur le document réponse les informations de l'entête conformément à cette illustration (exemple sur le fichier Cat.bmp):

```

> Magic number : BM
> Taille du fichier : 480054 octets
> Début de l'image : 54 octets
> Taille de l'entête : 40 octets
> Taille de l'image en pixels (X x Y) : 400x400 pixels
> Nombre de bit(s) pour coder un pixel : 24 bit(s)
> Méthode de compression : 0
> Taille de l'image : 480000 octets
> Nombre de couleurs importantes : 0 couleurs
> Résolution de l'image (X x Y) : 2834x2834 pixels par mètre

```

Exercice 2 :

a. Ecrire un nouveau programme qui permet d'afficher le contenu du fichier de la fin de l'entête jusqu'à la fin du fichier en précisant la position de chaque donnée conformément à cet exemple :

```
> 54 (0x36) : 0
> 55 (0x37) : 0
> 56 (0x38) : 255
> 57 (0x39) : 0
> 58 (0x3a) : 0
> 59 (0x3b) : 255
> 60 (0x3c) : 0
> 61 (0x3d) : 255
> 62 (0x3e) : 0
> 63 (0x3f) : 0
> 64 (0x40) : 255
> 65 (0x41) : 0
> 66 (0x42) : 0
> 67 (0x43) : 0
> ...
```

La structure du programme principal sera la suivante :

- Ouverture du fichier
- Lecture de la position du début de l'image (`Offset`)
- Parcourt le fichier en commençant à la position `Offset` et affiche les données.
- Fermeture du fichier

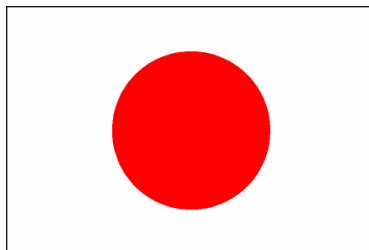
La fonction `int feof(FILE *stream)` permet de savoir si l'on atteint la fin du fichier. Relever sur le document réponse le contenu de l'image `BMP_2x2.bmp`.

Exercice 3 :

Ecrire un nouveau programme qui crée une image de 60x40 pixels en 16 millions de couleurs contenant le drapeau français. Utiliser la fonction `fwrite` qui permet d'écrire dans le fichier sur le même principe que `fread`.

Exercice supplémentaire :

Ecrire un nouveau programme qui crée une image de 60x40 pixels en 16 millions de couleurs contenant le drapeau japonais :



Compte-rendu à compléter

Exercice 1 :

a. Que signifie l'option "rb"

b. Explication des paramètres :

fread (MagicNumber,1,2,Img);	
MagicNumber	
1	
2	
Img	

fread (&FileSize,4,1,Img);	
&FileSize	
4	
1	
Img	

d. Taille relevée pour le fichier BMP_2x2.bmp : _____ octets

Explication :

Taille relevée pour le fichier IUT.bmp : _____ octets

Explication :

e. Entête pour le fichier BMP_2x2.bmp :

> Magic number : _____
> Taille du fichier : _____ octets
> Début de l'image : _____ octets
> Taille de l'entête : _____ octets
> Taille de l'image en pixels (X x Y) : _____ pixels
> Nombre de bit(s) pour coder un pixel : _____ bit(s)
> Méthode de compression : _____
> Taille de l'image : _____ octets
> Nombre de couleurs importantes : _____ couleurs
> Résolution de l'image (X x Y) : _____ pixels par mètre

Entête pour le fichier IUT.bmp :

> Magic number : _____
> Taille du fichier : _____ octets
> Début de l'image : _____ octets
> Taille de l'entête : _____ octets
> Taille de l'image en pixels (X x Y) : _____ pixels
> Nombre de bit(s) pour coder un pixel : _____ bit(s)
> Méthode de compression : _____
> Taille de l'image : _____ octets
> Nombre de couleurs importantes : _____ couleurs
> Résolution de l'image (X x Y) : _____ pixels par mètre

Exercice 2 :

a. Compléter et commenter le tableau suivant :

Position	Valeur	Commentaire
0x36	255	B=0xFF, V=0, R=0, c'est un pixel bleu.
0x37	0	
0x38	0	

Exercice 3 et supplémentaire :

A la fin de chaque exercice : commenter, indenter le programme et appeler l'enseignant afin qu'il valide votre code source.

Tableau à compléter par l'enseignant :

Exercice 3 :	Exercice supplémentaire :