TP4

Gilles Menez - UNS - UFR Sciences - Dépt. Informatique

14 octobre 2018

1 E/S avec la libC

Nous allons travailler (lecture, traitement, écriture) sur des images numériques.

Avec cette faible résolution, on voit bien les "pixels" de l'image.

1.1 Représentation des images numériques

Une telle image correspond à un tableau de $nx * ny$ pixels (Picture Element). Ces pixels sont le résultat d'un échantillonnage régulier des dimensions x et y.

➢ S'il s'agit d'une image couleur, un pixel est codé par 3 composantes (r,g,b) (chacune comprise au sens large entre 0 et 255), représentant respectivement les "doses" de rouge, vert et bleu qui caractérisent la couleur du pixel. Exemples de couleurs :

(0, 0, 0)	=	noir
(255, 0, 0)	=	rouge
(0,255, 0)	=	vert
(0, 0,255)	=	bleu
(127,127,127)	=	gris moyen
(255,255,255)	=	blanc

➢ S'il s'agit d'une image en niveau de gris, le pixel est codé par une composante comprise au sens large entre 0 et 255, représentant la luminosité du pixel.

0 pour noir,
ensuite (≥ 0) on va vers le blanc.

1.2 Visualisateurs d'images numériques

`gimp` est un excellent visualisateur/logiciel de traitement d'images.

Remarque : Ces visualisateurs permettent bien sûr de voir, mais aussi de générer des images au format souhaité en changeant de format lors de la sauvegarde.

1.3 Entrées-Sorties

Il existe une multitude de formats de stockage adaptés aux images : jpeg, gif tiff, png ...
Beaucoup mettent en œuvre des algorithmes de compression (réduction de la taille nécessaire au stockage) souvent très complexes.

Durant ce travail, nous utiliserons les formats "pgm" qui permettent de sauvegarder **simplement** dans des fichiers, car en format texte et sans compression des images.

Vous pouvez avoir plus d'informations sur ces formats en utilisant l'aide en ligne Unix :

➢ `man pgm` (Niveau de gris),

1.3.1 Format d'un fichier `pgm`

Si vous lisez un fichier `pgm` avec `emacs` ou juste avec un `more`, vous obtenez :

```
P2
# CREATOR: XV Version 3.10a  Rev: 12/29/94 (PNG patch 1.2)
# CREATOR: The GIMP's PNM Filter Version 1.0
14 8
255
  0   7  25  43  61  79  96 114 132 150 168 186 204 221   0  10  28
 45  63  81  99 117 135 152 170 188 206 224   0  12  30  48  66  84
102 119 137 155 173 191 209 226   0  15  33  51  68  86 104 122 140
158 175 193 211 229   0  17  35  53  71  89 107 124 142 160 178 196
214 232   2  20  38  56  73  91 109 127 145 163 181 198 216 234   5
 22  40  58  76  94 112 130 147 165 183 201 219 237   7  25  43  61
 79  96 114 132 150 168 186 204 221 239
```

Vous pouvez constater que ces fichiers **contiennent deux parties** :

① L'en-tête :

```
P2
# CREATOR: XV Version 3.10a  Rev: 12/29/94 (PNG patch 1.2)
# CREATOR: The GIMP's PNM Filter Version 1.0
14 8
255
```

Il comporte plus précisément :

➢ un nombre magique : $P2$
Cette chaîne de caractères identifie le type du fichier.

➢ le nombre de colonnes (nx) et le nombre de lignes (ny)

➢ le nombre de niveaux de gris de l'image qui sont nécessaires au logiciel de visualisation qui va afficher l'image : 255 veut dire qu'il peut y avoir 256 niveaux de gris, de 0 à 255, chaque valeur correspondant à un gris différent.

Les lignes qui commencent par le caractère `#` sont des lignes de commentaires qui sont ignorées par le logiciel de visualisation.

② L'information : c'est-à-dire les points de l'image dont la valeur est un entier.
Dans l'exemple ($nx = 14$ et $ny = 8$) :

`I[0][0] ..... I[0][13] I[1][0] ..... I[7][13].`

Le traitement qui va vous être demandé tolère que l'on représente l'image dans un tableau 1D. Il est d'ailleurs fréquent qu'une image soit représenté par un tableau 1D quelque soit le traitement. C'est juste un problème d'indexation ...

2 Votre travail

2.1 E/S de haut niveau (libc)

Dans un premier temps, vous allez permettre au fichier `main.c` qui vous est donné (sur le site Web) de réaliser ses actions.

En conservant les choix de typage faits dans le fichier `image.h` (lui aussi est donné), vous proposez 3 fonctions dans un fichier `image.c` :

① Lire un fichier contenant une image au format "pgm" (ASCII)

② Inverser la valeur de chaque pixel, en faisant en sorte que le noir devienne le blanc.

$$\text{new\_pixel} = \text{max\_level} - \text{old\_pixel}$$

③ Ecrire un fichier conforme au format "pgm".

➢ Le fichier doit être visualisable avec un outil de type `gimp` .

En ce qui concerne les commentaires, on peut les supprimer ou pas si vous savez faire ?

Vous avez un exemple d'exécution à la fin du sujet.

2.2 Exécutable et bibliothèque

Vous devez illustrer l'exécution de votre programme de trois façon :

① **Classique** : vous compilez les deux fichiers sources simultanément.
Vous obtenez un exécutable `image` .

② **Bibliothèque statique** : vous faîtes de `image.c` une bibliothèque statique que vous placez dans un répertoire `lib` .

Le programme est obtenu en liant cette bibliothèque statique du sous répertoire `lib` .

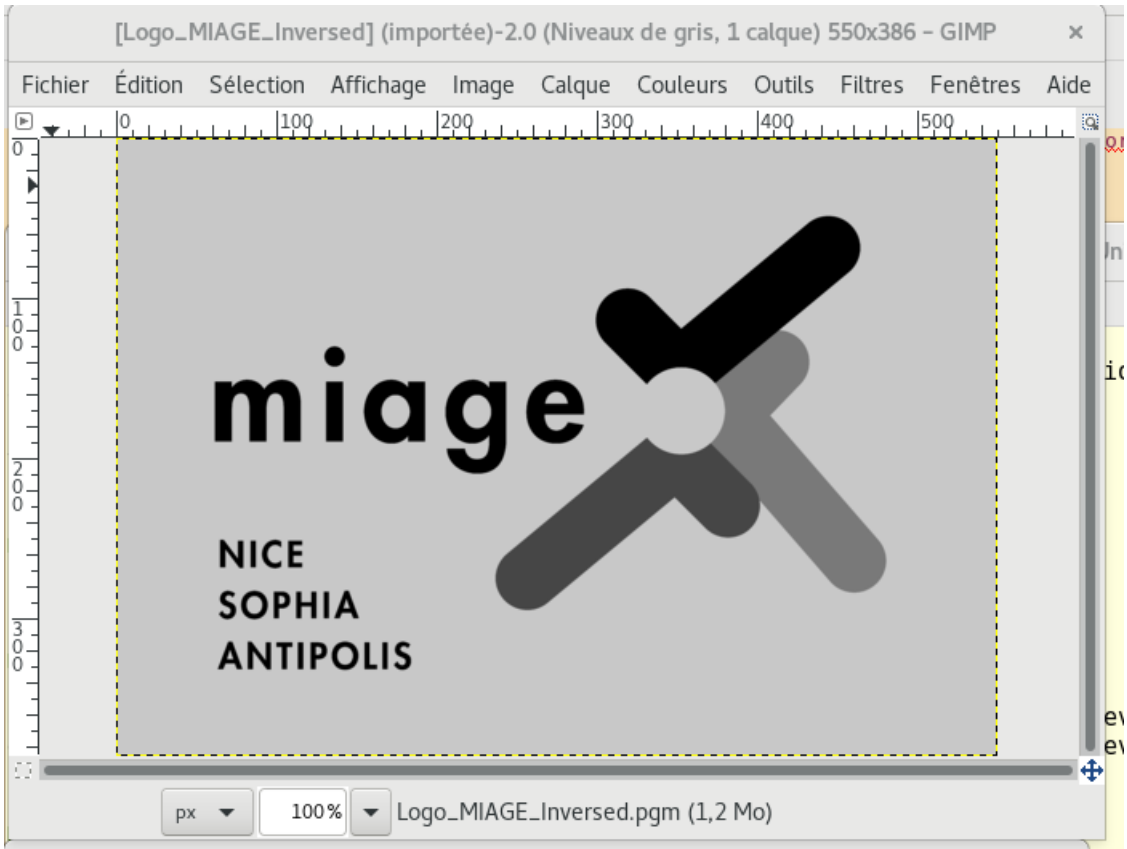
L'exécutable se nomme `image_static` .

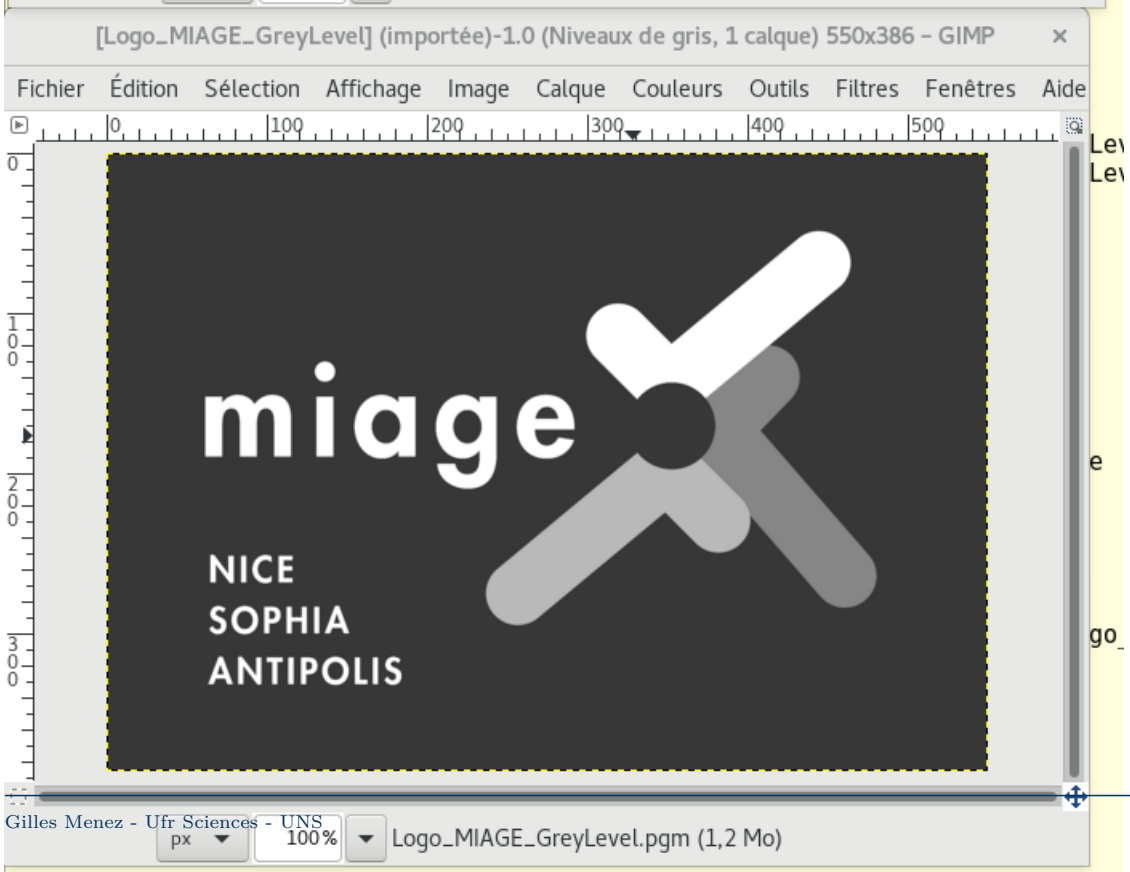
③ **Bibliothèque dynamique et partagée** : vous faîtes de `image.c` une bibliothèque "shared" (toujours dans le répertoire `lib`).

L'exécutable lié avec cette bibliothèque se nomme `image_dyn` .

2.3 Analyse

① Faire marcher.
② Comparer la taille des trois exécutables.
③ Montrer le placement en mémoire de la bibliothèque partagée.
④ Montrer les effets des allocations dans le tas dans la mémoire du processus.






```
menez@mowgli ~/EnseignementsCurrent/Cours_Sys_Prog/Tps/TP4_Unix/Src
$ ./image Logo_MIAGE_GreyLevel.pgm Logo_MIAGE_Inversed.pgm
Done !
```

2.4 Plugin

Pour ceux qui se sont "baladés" ...

① Vous créez un nouveau fichier imageplus.c

② Vous placez dans ce fichier une nouvelle fonction de traitement de l'image.

void image_NB(image *, int seuil) ;

Cette fonction rend une image en noir et blanc :

➢ Les pixels au dessus du seuil valent le niveau max.

➢ Les pixels au dessous du seuil valent 0.

③ En utilisant le tutorial :
`http://www.yolinux.com/TUTORIALS/LibraryArchives-StaticAndDynamic.html`
et plus précisément la partie : "Dynamic loading and un-loading of shared libraries using libdl"

faire découvrir cette nouvelle fonction à votre programme.