

INF304 — TP1

Première partie : arguments de la ligne de commande et fichiers en C

Préambule

Les TP se font sur le serveur **turing** (environnement Unix). Pour le premier TP, suivre la procédure de connexion et de mise en place des fichiers de configuration [sur cette page](#).

Avant de commencer ce premier TP vous devez créer un répertoire de nom INF304 (`mkdir INF304`). Placez-vous alors dans ce répertoire (`cd INF304`), et copiez les fichiers nécessaires à ce premier TP :

```
cp -r /Public/304_INF_Public/TP1 .
```

N'oubliez pas le "." à la fin de cette commande !

Vous pouvez alors vous placer dans le répertoire TP1 qui vient d'être créé (`cd TP1`), puis dans le sous-répertoire TP1a.

Arguments de la ligne de commande en C

L'accès aux arguments de la ligne de commande depuis un programme C se fait en utilisant les deux paramètres `argc` et `argv` de la procédure principale `main` :

- `argc` contient le nombre d'arguments de la ligne de commande (nom du programme inclus) ;
- `argv` est un tableau de chaînes de caractères (une chaîne de caractère étant représentée en C par un pointeur de caractère) : `argv[i]` est l'argument n°i. `argv[0]` contient le nom du programme exécuté.

Pour mieux comprendre le fonctionnement de ces fonctions :

1. Lisez le code du programme `essai_ligne_de_commande.c` en utilisant un éditeur de texte (emacs, nedit ou gedit).
2. Compilez ce programme avec la commande `clang essai_ligne_de_commande.c -o essai_ligne_de_commande`.
3. Exécutez-le avec différentes séquences d'arguments, comme par exemple :

```
./essai_ligne_de_commande abc 42 defg  
./essai_ligne_de_commande  
./essai_ligne_de_commande abc
```

Exercice 1

Écrire un programme C de nom `exercice1.c` qui affiche un message d'erreur s'il ne reçoit pas exactement deux arguments sur sa ligne de commande, et qui affiche ces deux arguments dans le cas contraire. Compilez ce programme (`clang`

`exercice1.c -o exercice1`) et testez-le.

Lecture et écriture de «fichiers texte» depuis un programme C

Référez-vous à la partie «Lecture et écriture de fichiers texte depuis un programme C» du Mémento C.

Exercice 2

Écrire un programme C de nom `exercice2.c` qui lit la première ligne d'un fichier et copie son contenu dans un autre fichier. Les noms des deux fichiers seront passés en argument de la ligne de commande, un message d'erreur sera affiché si la ligne de commande est incorrecte. Compilez ce programme et testez-le.

Exercice 3

Écrire un programme C de nom `exercice3.c` qui lit un fichier caractère par caractère et copie son contenu dans un autre fichier. Les noms des deux fichiers seront passés en argument de la ligne de commande, un message d'erreur sera affiché si la ligne de commande est incorrecte.

Compilez et testez ce programme.

Exercice 4

1. Écrire un programme C de nom `exercice4.c` qui lit dans un fichier (dont le nom est fourni sur la ligne de commande) une date au format "jour mois année" (comme 18 octobre 1966) avec le jour sur la première ligne, le mois sur la deuxième ligne et l'année sur la troisième, et affiche un entier représentant la somme du jour et de l'année (comme 1984, somme de 18 et 1966).

Exemple de fichier :

```
18
octobre
1966
```

Compilez et testez ce programme.

2. Modifier le programme pour lire une date dans un fichier avec le jour, le mois et l'année sur la même ligne.

Avant et après le nom du mois, il y a (au moins) un espace ou tabulation.

Exemple de fichier :

```
18 octobre 1966
```

Deuxième partie : transformations d'images

Placez-vous dans votre répertoire INF304/TP1/TP1b.

Le but de ce TP est la manipulation d'images au format PGM Texte. Les images sont des rectangles de largeur L et de hauteur H, constitués de H lignes, chaque ligne contenant L pixels.

Un pixel est défini par son intensité, un entier compris entre 0 (si le pixel est noir) et max (si le pixel est blanc), une valeur intermédiaire correspondant à un gris plus ou moins foncé. max est un entier entre 1 et 255 (255 en général).


Un fichier au format PGM Texte est un fichier texte constitué comme suit :

- la première ligne contient la chaîne P2.
- la deuxième ligne commence par le caractère # (il s'agit d'un commentaire).
- puis, séparés par des espaces ou des retours à la ligne, les entiers L, H et max.
- enfin, séparés par des espaces ou des retours à la ligne, les H lignes de L valeurs des pixels (comprises entre 0 et max).

Votre répertoire contient quelques images dans ce format (ce sont les fichiers dont le suffixe est .pgm).

Exercice 1

1. Visualisez (à l'aide de l'éditeur ou de la commande `more` du terminal) le contenu du fichier `INF304.pgm` (par exemple).

En parallèle, ouvrez le même fichier image. Vous pouvez pour cela installer l'extension « PBM/PPM/PGM Viewer » sous Visual Studio (bouton extension  ; rechercher « PGM » ; installer l'extension), il suffit ensuite de cliquer sur le fichier image dans l'explorateur de fichiers.

Vérifiez que vous comprenez bien le format des fichiers.

2. L'ensemble du programme fourni est réparti entre différents fichiers. Repérez ce que contient chacun d'eux.
3. Compilez le programme à l'aide de la commande `make` et exécutez-le. Quel est l'effet du programme (en cas de «bonne utilisation») ?

Exercice 2

Modifiez/complétez la fonction inverse video afin qu'elle respecte sa spécification, et testez.

Exercice 3

Complétez le paquetage `Operations_image` avec une ou plusieurs transformation(s) de votre choix. Testez !

Compte-rendu

À l'issue du TP, déposer sur [la page moodle du cours](#) :

- Un compte-rendu d'une page maximum, au format PDF :
 - Pour chaque paquetage du programme initialement fourni, décrivez son contenu en une phrase.

- Que fait le programme initialement fourni ?
- Décrivez en une phrase, chacune des transformations que vous avez ajoutée.
- Le fichier `operations_image.c` modifié.

>