Pratique de la programmation
TP 1 : arguments de programmes et fichiers

Frédéric Vivien, e-mail : vivien@icps.u-strasbg.fr

Ce TP a pour but la manipulation des fichiers et le passage d'arguments à un programme via les paramètres `argc` et `argv`. Les différents programmes et les différents fonctions devront être testés au fur et à mesure de leur écriture. Si vous rencontrez des problèmes de syntaxe, ou avez des doutes sur le comportement exact d'une fonction, pensez à consulter le `man` !

1. Écrivez un programme prenant en entrée un nombre quelconque d'arguments. Tous les arguments seront des noms de fichiers, à l'exception du premier argument, qui sera le nombre des fichiers passés en argument.

   Le programme devra vérifier que son premier argument est correct, et devra afficher les noms des différents fichiers.

   On pourra utiliser la fonction `atoi`.
2. Même question que précédemment, le programme devant en plus ouvrir en lecture seule tous les fichiers reçus en argument, avant de les refermer. Le programme devra afficher si ces fichiers existent ou non.
3. Écrivez une fonction qui prend en entrée un nom de fichier, et qui copie le contenu du fichier à l'écran.
4. Écrivez une fonction qui prend en entrée un nom de fichier source et un nom de fichier destination, et qui recopie le fichier source dans le fichier destination. Pour ce faire on utilisera les fonctions `fgets` et `fputs`. On pourra contrôler la bonne exécution de cette fonction au moyen de la fonction `diff` (hors du programme, dans un terminal).
5. Même question qu'en 4, les lectures et écritures se faisant cette fois au moyen des fonctions `fread` et `fwrite`.
6. Écrivez un programme qui prend en entrée : le nombre de fichiers qu'il doit lire, les noms de ces fichiers, et le nom du fichier de sortie, et qui recopie le contenu de tous les fichiers sources dans le fichier destination.
7. Même question qu'en 6, mais en refermant le fichier destination entre la copie de deux fichiers sources.
8. Nous considérons maintenant des fichiers contenant des informations structurées de la forme suivante :

   ```
   Nom Prénom
   JJ/MM/YYYY
   Niveau
   ```

   où le `Nom` et le `Prénom` sont toujours codés au moyen de 24 caractères chacun, et où `niveau` est codé au moyen de 8 caractères (`IUP1    `, `Maîtrise`, etc).

   (a) Écrivez une fonction qui recopie à l'écran le contenu d'un tel fichier, chaque structure étant lu grâce à un seul appel à la fonction `fscanf`.

   (b) Écrivez une fonction qui prend en entrée un entier `i` et un `niveau`, et qui réécrit le `niveau` de la n^{e} structure. Pour ce faire on utilisera la fonction `fseek`.

   (c) Même question qu'en 8a, la recopie ce faisant cette fois-ci dans un fichier.