

TP n°1

Règles du jeu :

- on écrit en langage C uniquement (donc on compile avec **gcc**)
- à propos des appels système et autres fonctions standards, nous avons le droit à :
 - **write** et **read** (dans *unistd.h*). Leur doc est dans la section 2 du man.
 - **open** et **close** (dans *fcntl.h*). Leur doc est dans la section 2 du man.

Exercice 1

Écrire un programme **hello** qui affiche le texte **Hello world** sur la sortie standard (console).

Exercice 2

Écrire un programme **monCat** qui lit une ligne de texte entrée au clavier puis affiche cette ligne sur la sortie standard. Pour cela, vous stockerez dans un tableau statique (avec une taille fixée, ex : 256 caractères) ce que l'utilisateur rentre dans l'entrée standard, et vous détecterez lorsque l'utilisateur effectue un retour à la ligne (caractère '\n').

Exercice 3

Écrire un programme **monEcho** qui simule le comportement de la commande unix echo. Par exemple, l'exécution de

```
./monEcho coucou comment          allez          vous ?
```

aura pour effet d'afficher :

```
coucou comment allez vous ?
```

Indication : il suffit de parcourir le tableau **argv**, et connaître la taille de chaque chaîne. Pour cela, vous détecterez la fin de chaque chaîne grâce au symbole '\0'.

Exercice 4

Écrire un programme **monCp** qui prend en arguments deux noms de fichiers (supposés contenir du texte) et copie le contenu du premier fichier dans le second. Pour cela on utilisera un tampon (buffer) initialisé à 1000, c'est à dire que le programme lira 1000 caractères du premier fichier, les copiera dans le second, puis lira encore 1000 caractère du premier fichier, et ainsi de suite jusqu'à atteindre la fin du premier fichier.

Votre programme affichera des messages d'erreur appropriés sur la sortie d'erreur standard dans les cas suivants :

- nombre d'arguments différent de 2,
- problème à l'ouverture d'un des deux fichiers,
- problème à l'écriture.

Pour traiter ces messages d'erreur, vous utiliserez la fonction **perror** de **stdio.h**, qui utilise la variable globale **errno** définie dans **errno.h**. Vous aurez ainsi à lire la documentation du man (ou bien d'autres ressources si vous le souhaitez).

Si le deuxième nom passé en argument correspond au nom d'un fichier existant, alors le contenu de ce fichier sera écrasé ; sinon vous devrez créer ce fichier (il devra avoir les droits en lecture et en écriture pour l'utilisateur).

Bonus Exercice 4

Écrire un programme monCp2 qui est une copie du programme précédent, mais utilise un tampon d'un seul caractère. Comparer l'efficacité des deux programmes sur des fichiers contenant peu de texte et beaucoup de texte, et essayez de comprendre le résultat.