

Appels Système

open, read, write, close

Nb : Le TP manipule les les appels système. Vous ne pouvez pas utiliser la fonctions sur les fichiers de la librairie standard : `fopen, fread, fwrite ...`

Copie de fichier

Ecrire une commande *copy* qui effectue une copie d'un fichier dans un autre en utilisant les appels système unix (*open, read, write, close*. Les noms des deux fichiers sont passés en arguments de la commande. La commande doit travailler sur des blocs de 512 caractères.

Le plus simple est de repartir des deux programmes donnés en cours : Cat et Tee.

Masjuscule

- Ecrire une commande *upper* qui copie le contenu d'un fichier en remplaçant les lettres minuscules en majuscules. Les noms des fichiers source et résultat sont passés en argument de la commande. Vous devez pour cela utiliser les appels système unix (*open, read, write, close*).

Vous pouvez utilisez la librairie `/usr/include/ctype.h` avec *islower* et *toupper* ou (En python, plus simplement la méthode `.upper()`). Les caractères sont lus un par un dans le fichier (pas de buffer). On suppose que le fichier en paramètre est un fichier texte (un octet par caractère , pas d'encodage UTF8)

- Même question, mais cette fois ci, si un seul nom de fichier est passé en argument de la commande, vous devez utiliser la sortie standard pour le résultat.
- Même question, mais cette fois ci, si aucun nom de fichier est passé en argument de la commande, vous devez utiliser l'entrée standard et la sortie standard.

Concatennation

Ecrire une commande *concat* qui concatenne plusieurs fichiers dans un fichier résultat. La liste des noms des fichiers est passée en argument de la commande le dernier étant le nom du fichier résultat. Si un des fichiers n'existe pas, la commande passe au fichier suivant. Les fichiers sont lus par blocs de 512.