

TP Système Exploitation

Révision I/O

2022

Exercice 1: Réalisation du programme ‘tee’

La commande `tee` effectue des lectures sur son entrée standard et recopie les octets lus dans la sortie standard *et* dans un fichier spécifié en argument.

Etape 1: le programme de base

Proposez un programme en C nommé `montee` qui réalise la même fonction que `tee` en utilisant uniquement les appels E/S de bas niveau.

Etape 1: l’option -a

Par défaut, la commande `tee` écrase le contenu du fichier existant.

Par contre si la ligne de commande comporte l’option “-a” alors le contenu est ajouté à la fin du fichier existant.

Réalisez cette option dans votre programme `montee`.

Exercice 2: Réalisation d’un éditeur de fichier par ligne de commande

L’objectif de cet exercice est de réaliser un éditeur de fichier en utilisant uniquement la ligne de commande.

Votre programme recevra comme premier paramètre le path du fichier à éditer et une suite de commandes.

Voici la liste des commandes que votre programme devra interpréter :

- **s<decalager>** : réaliser un décalage du curseur de valeur **<decalage>** à partir du début du fichier
- **r<longueur>** : lecture de **<longueur>** octets à partir de la position actuelle du curseur et affichage de ces octets sur la sortie standard
- **w<str>** : écriture de la suite de caractères **<str>** dans le fichier à partir de la position actuelle du curseur.

Pour réaliser cet exercice vous devez utiliser les fonctions d'accès E/S bas niveau (**open**, **read**, **write**, **close** et **lseek**)

Exemple d'utilisation

Voici un exemple d'utilisation. Notre programme est nommé **editor**

```
$ touch monfichier
$ stat -l monfichier
-rw-r--r-- 1 x x 0 monfichier
$ ./editor ./monfichier s256 wCouCou
s256: déplacement du curseur réussi
wCouCou: écriture de 6 bytes
$ stat -l monfichier
-rw-r--r-- 1 x x 262 monfichier
$ ./editor ./monfichier s252 r10
s252: déplacement du curseur réussi
r10: 00 00 00 00 CouCou
$
```

On remarque qu'après la création avec la commande **touch** le fichier à 0 octets comme taille (retour de commande **stat**)

L'éditeur effectue un déplacement de 256 octets et ensuite écrit le mot 'CouCou'

La commande **stat** cette fois-ci nous donne la taille du fichier qui est de 262 octets (=256 + 6). Il faut remarquer que c'est un fichier qui comporte un 'trou' de 256 octets.

La dernière ligne demande un déplacement du curseur à l'octet 252 et une lecture de 10 octets.

On affiche bien 6 valeurs null à cause du trou (00 en hexa) et les caractères de la chaîne CouCou

Fonctions utiles pour réaliser le programme:

Voici la liste des fonctions C utiles pour la réalisation du programme demandé: **open**, **close**, **write**, **read**, **lseek**, **strtol**, **malloc**, **printf**, **isprint**

Merci de consulter la documentation de ces fonctions avant de les utiliser.