

# TD Programmation système et réseaux

## No 4

Avril 2003

### Rappel

Headers à inclure :

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <errno.h>
```

Toutes les fonctions qui suivent rendent -1 en cas d'erreur, la variable `errno ()` contient le code de l'erreur

- Ouverture d'un fichier : `int open(char *nom_f, int mode, .../*mode_t droits*/);`
  - `mode` : `O_RDONLY`, `O_WRONLY`, `O_RDWR`, `O_CREAT`, `O_TRUNC` (si `nom_f` existe, destruction), `O_APPEND`
  - `droits` : `S_IRUSR`, ..., `S_IWGRP`, ..., `S_IXOTH`.
  - Rend un descripteur de fichier
- Création d'un fichier : `int creat(char nom_f, mode_t droits);` équivalent à `open(nom_f, O_WRONLY|O_CREAT|O_TRUNC, droits)`.
  - Rend un descripteur de fichier.
- Fermeture d'un fichier : `int close(int desc);`
  - Rend 0 si succès
- Lecture dans un fichier : `int read(int desc, void *ptr, int nb_octets);`
  - renvoie le nombre d'octets réellement lus
- Ecriture dans un fichier : `int write(int desc, void *ptr, int nb_octets);`
  - renvoie le nombre d'octets réellement écrits
- \* Déplacement dans un fichier : `int lseek(int desc, int offset, int origine);`
  - `origine` = `SEEK_SET`, `SEEK_CUR`, `SEEK_END`.
  - Renvoie le nombre d'octets entre le premier octet du fichier et la position courante.
- Suppression d'un fichier : `int unlink(const char *pathname);`
  - rend 0 si succès, la suppression n'est réelle dans le système de fichiers que si le nombre de liens tombe à zéro

### Exercices

1. Ecrire un programme qui lit un fichier et écrit son contenu sur la sortie standard en transformant les majuscules en minuscules. On pourra utiliser la fonction `char tolower(char)` qui réalise cette conversion (renvoie le caractère inchangé s'il ne s'agit pas d'une majuscule). Cette fonction demande `#include <sys/ctype.h>`
2. Ecrire un programme qui concatène une suite de fichiers donné en paramètre sur la ligne de commande (commande `cat`).

3. Idem pour une version simplifiée de la commande cp ne fonctionnant que pour 2 fichiers réguliers.
4. Soit les fichiers toto et titi créés respectivement par le programme C et la commande bash figurant dans le tableau ci-dessous :

```
main() {  
    int d, val=1234567;  
    d = open("toto", O_RDWR|O_CREAT, S_IRWXU);  
    if (d == -1) exit(1);  
    write(d, &val, sizeof(int));  
    close(d);  
}
```

et la commande bash suivante : echo 1234567 > titi

- (a) Quelles seront les tailles de ces fichiers ?
  - (b) Sachant qu'ils contiennent chacun un entier, comment peut t-on retrouver la valeur des entiers ?
5. Ecrire un programme C qui donne la taille d'un fichier.
  6. Ecrire la commande wc.