

TD de Système n° 10

Exercice 1 : manipulation des descripteurs et dup()

On considère le programme `testdup.c` ci-contre ; indiquer l'effet de la suite d'instructions suivante :

```
$ gcc -Wall -o testdup testdup.c
$ echo '01234567890123456789' > tutu
$ ./testdup tutu ; cat tutu
```

Exercice 2 : dup2()

Écrire une fonction `void cat()` qui copie l'entrée standard sur la sortie standard.

Écrire un programme utilisant cette fonction qui copie le fichier dont le nom est donné dans `argv[1]` sur celui dont le nom est dans `argv[2]` (en utilisant l'appel système `int dup2(int oldfd, int newfd);`).

Modifier ce programme pour afficher « Copie terminée » sur la sortie standard après le retour de `cat()`.

Exercice 3 : Enchaîner l'exécution

Écrire une commande `./pipe prog1 prog2` simulant le comportement de `prog1 | prog2` dans un shell, c'est-à-dire exécutant *simultanément* les deux programmes, redirigeant la sortie standard du premier dans l'entrée standard du deuxième, et retournant le code de retour du deuxième.

```
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>

void erreur(const char *mess) {
    perror(mess); exit(2);
}

void usage(const char *s) {
    fprintf(stderr, "Usage : %s reference\n", s); exit(1);
}

int main(int argc, char *argv[]) {
    int d1, d2, d3;
    char buf[3];

    if (argc != 2) usage(argv[0]);
    if ((d1 = open(argv[1], O_RDONLY)) == -1) erreur("open 1");
    if ((d2 = open(argv[1], O_WRONLY)) == -1) erreur("open 2");
    if ((d3 = open(argv[1], O_RDWR)) == -1) erreur("open 3");
    buf[read(d1, buf, 2)] = '\0';
    printf("lu sur d1 : %s\n", buf);
    write(d2, "abc", 3);
    buf[read(d3, buf, 2)] = '\0';
    printf("lu sur d3 : %s\n", buf);
    write(d3, "xxxx", 4);

    if ((d1 = dup(d3)) == -1) erreur ("dup 1");
    if ((d2 = dup(d3)) == -1) erreur ("dup 2");
    buf[read(d1, buf, 2)] = '\0';
    printf("lu sur d1 : %s\n", buf);
    write(d2, "abc", 3);
    buf[read(d3, buf, 2)] = '\0';
    printf("lu sur d3 : %s\n", buf);
    write(d3, "xxxx", 4);

    exit(EXIT_SUCCESS);
}
```