Exercice :

Écrire un programme en langage C qui simule le comportement de la commande **ls -al | wc -l** en utilisant deux processus qui communiquent à travers un tube. Le processus parent exécutera ls -al et enverra sa sortie au processus enfant, qui exécutera **wc -l** pour compter le nombre total de lignes.

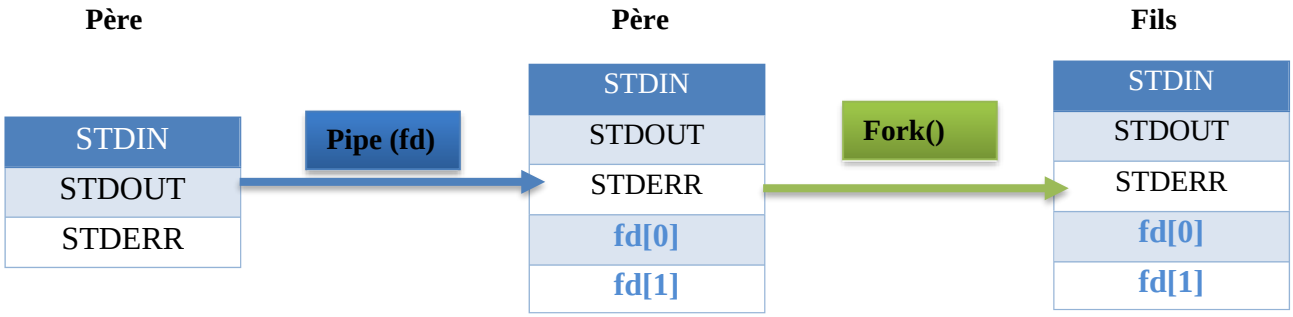
Pour réaliser cela, le processus parent devra rediriger sa sortie standard vers l'entrée du tube avant d'exécuter ls -al, tandis que le processus enfant devra rediriger son entrée standard depuis le tube avant d'exécuter **wc -l**. Utilisez la primitive **execlp** pour exécuter les commandes **ls** et **wc respectivement.**

Assurez-vous que votre programme produit le même résultat que **ls -al | wc -l**, c'est-à-dire le nombre total de fichiers et de répertoires répertoriés dans le répertoire courant.

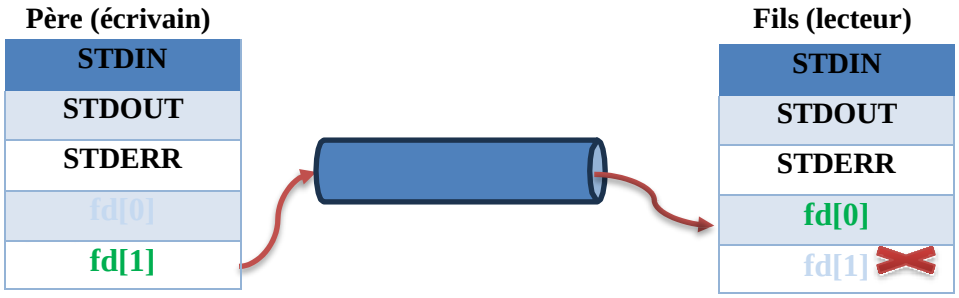
ls -al : ***La commande ls -al est une commande utilisée dans les systèmes Unix et Linux pour afficher une liste des fichiers et répertoires dans un répertoire donné, ainsi que leurs détails.***

wc -l: ***La commande wc -l est une commande utilisée dans les systèmes Unix et Linux pour compter le nombre de lignes dans un flux d'entrée donné.***

Création



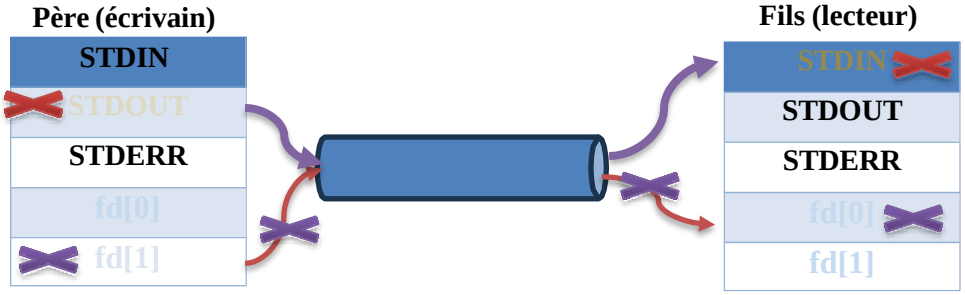
Communication



Close (fd[0])

Close fd[1])

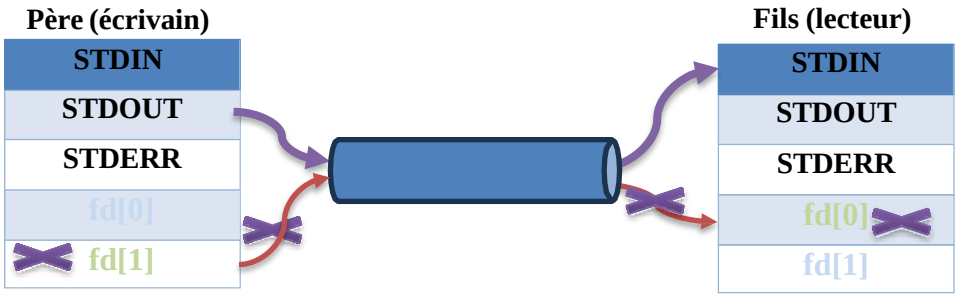
Redirection (dup)



Close (STDOUT)
Dup(fd[1])
Close (fd[1])
execlp (“ls”, “ls”, “-al”, NULL)

Close (STDIN)
Dup(fd[0])
Close fd[0])
execlp (“wc”, “wc”, “-l”, NULL)

Redirection (dup2)



Dup2(fd[1], STDOUT)
Close (fd[1])
execlp (“ls”, “ls”, “-al”, NULL)

Dup2(fd[0], STDIN)
Close fd[0])
execlp (“wc”, “wc”, “-l”, NULL)

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>

int main() {
    int fd[2]; // Tableau pour les descripteurs de fichier du tube

    // Création du tube
    pipe(fd);

    pid_t pid = fork();
    if (pid < 0) {
        perror("Erreur lors de la creation du fork");
    }

    if (pid == 0) { // Processus enfant
        // Rediriger l'entrée standard vers l'extrémité de lecture du tube
        close(fd[1]); // Fermer l'extrémité d'écriture du tube
        dup2(fd[0], STDIN_FILENO);
        close(fd[0]); // Fermer l'extrémité originale après duplication

        // Exécuter la commande wc -l
        execlp("wc", "wc", "-l", NULL);

    } else { // Processus parent
        // Rediriger la sortie standard vers l'extrémité d'écriture du tube
        close(fd[0]); // Fermer l'extrémité de lecture du tube
       dup2(fd[1], STDOUT_FILENO) ;

        close(fd[1]); // Fermer l'extrémité originale après duplication

        // Exécuter la commande ls -al
        execlp("ls", "ls", "-al", NULL);
    }

    // Le processus parent attend la fin du processus enfant
   wait(NULL) ;


    return 0;
}
```