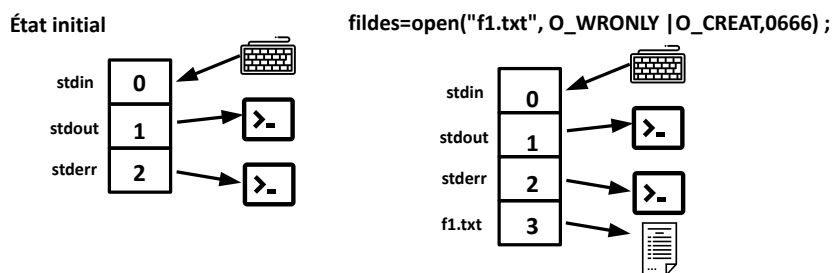


TP 2-2: redirections, fichiers



Rappel sur les descripteurs de fichiers :

Tous les processus Linux possèdent automatiquement 3 descripteurs, stdin (entrée standard clavier), stdout (sortie standard sur le terminal), stderr (sortie d'erreur standard sur le terminal, non bufferisée), de numéros respectifs 0,1 et 2. Ils sont manipulables par les primitives classiques des fichiers (`read`, `write`, `close`...), et il n'est pas nécessaire de les ouvrir par `open` (c'est déjà fait !). Quand un nouveau fichier est ouvert par `open`, il obtient le premier descripteur libre (par exemple, 3 au lancement du processus).



Exercice 1. Redirection

- Ecrivez le programme `mycat` qui, appelé sur le terminal sans arguments, reproduit le comportement de la commande `shellcat`. C'est-à-dire qu'il duplique sur la *sortie standard* (identifié par la constante `STDOUT_FILENO`, (égale en fait à 1)) les caractères qui sont entrés au clavier (*entrée standard* `STDIN_FILENO`) :

```
1 % cat
2 je copie
3 je copie
4 ^D
```

Vous utiliserez les primitives systèmes suivantes : `open`, `read`, `write`, `close` (et `perror` pour gérer les erreurs).

- Modifiez le programme pour qu'il accepte de façon optionnelle le nom d'un fichier (ex : *mycat fichier.txt*) en paramètre. Cette fois, le comportement sera celui de *cat > fichier.txt*.

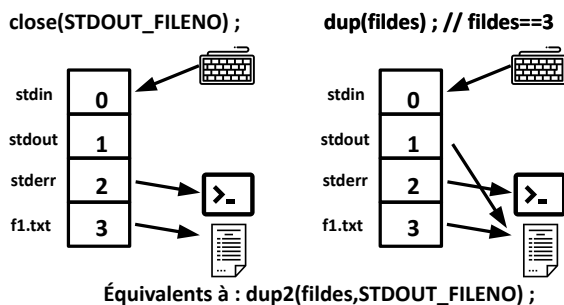
```
1 % cat > fichier.txt
2 je copie
3 ^D
4
5 % more fichier.txt
6 je copie
```

Vous devrez obligatoirement utiliser pour l'écriture (sur fichier comme sur la sortie standard) l'instruction suivante :

```
write(STDOUT_FILENO, buf, n)
```

Donc, dans le cas où un nom de fichier en argument est donné, il vous faudra **rediriger la sortie standard vers ce fichier**.

Vous utiliserez les primitives systèmes suivantes : *open*, *read*, *write*, *close* (et *perror* pour gérer les erreurs) ainsi que *dup*. On rappelle que cette instruction duplique un descripteur passé en paramètre dans le descripteur libre de plus petit numéro (*i.e.* le premier en partant de 0).



- Modifiez le programme pour utiliser **dup2** à la place de *dup*.

Exercice 2. fonctions standard vs systèmes

Le code fourni dans le fichier *partage_pere_fils_std_systeme.c* reprend l'exemple vu dans le cours2 (diapos 24-25). Vérifiez que le fonctionnement est bien celui décrit en cours. si vous relancez le programme, vous aurez un problème. Donnez une explication et proposez un correctif.