

Nous allons à présent ajouter à notre shell `iutsh` la gestion des tubes. N'oubliez pas de sauvegarder la version 1 avant de commencer les modifications.

La « gestion des tubes » consiste à traiter les lignes de commandes composées de plusieurs commandes en pipeline. Par exemple, la ligne

```
iutsh$ cat /etc/services | grep tcp | wc -l
```

permet de lancer `cat /etc/services` dont la sortie standard est redirigée vers un premier tube, puis la commande `grep tcp` dont l'entrée standard est redirigée vers ce premier tube et la sortie standard vers un second tube, et enfin la commande `wc -l` dont l'entrée standard est le second tube. Notez que d'une manière générale, pour exécuter une ligne de commande de la forme

```
iutsh$ commande1 | commande2 | ... | commande N
```

chaque commande est exécutée de manière « classique » après redirection de son entrée et de sa sortie standard (sauf pour la première qui garde l'entrée standard par défaut et la dernière qui garde la sortie standard par défaut). La première fonction va donc nous permettre d'exécuter une commande.

### Exercice 1 : Lancer une commande

Écrire la fonction

```
int lance_commande( int in, int out, char *com, char ** argv);
```

qui permet d'exécuter la commande `com` avec le tableau d'arguments `argv`. Un processus doit être créé pour exécuter cette commande, celui-ci doit utiliser respectivement les descripteurs `in` et `out` comme entrée et sortie standard. **Attention**, les descripteurs `in` et `out` peuvent éventuellement être 0 ou 1, dans ce cas il n'y a pas de redirection à effectuer.

La fonction retourne le `pid` du processus créé en cas de succès et `-1` en cas d'échec.

### Exercice 2 : Exécuter une ligne de commandes

Il s'agit à présent de modifier la fonction

```
void execute_ligne_commande (void);
```

réalisée lors de l'étape I. Au lieu de lancer une seule commande, la fonction lance successivement toutes les commandes analysées sur la ligne de commande en utilisant la fonction `lance_commande`.

**Rappel** : la fonction `ligne_commande` retourne un tableau de commande à lancer et indique par le paramètre `nb` le nombre de commandes du tableau.

Il suffit donc de lancer une à une les commandes saisies grâce à la fonction `ligne_commandes` en redirigeant les entrées/sorties vers les tubes adéquats. Avant de lancer chaque nouvelle commande, il est nécessaire de créer un tube et d'indiquer à la fonction `lance_commande` les descripteurs à utiliser pour cette commande.

**Attention** aux points suivants :

- la première et la dernière commande sont légèrement différentes, la première garde l'entrée standard « habituelle » et la dernière garde la sortie standard « habituelle » ;
- chaque fois que le processus « père » crée un tube, il y « connecte » un fils puis le suivant, ensuite, ce tube ne lui étend plus d'aucune utilité, il doit le fermer ;
- lorsqu'une erreur survient dans le lancement d'une commande, il faut continuer à exécuter les autres (essayez par exemple `ls | tralala | ls` en `bash`) ;
- lorsque la ligne de commandes est lancée en avant plan, votre `shell` doit attendre la fin de tous les processus qui ont été créés avec succès.