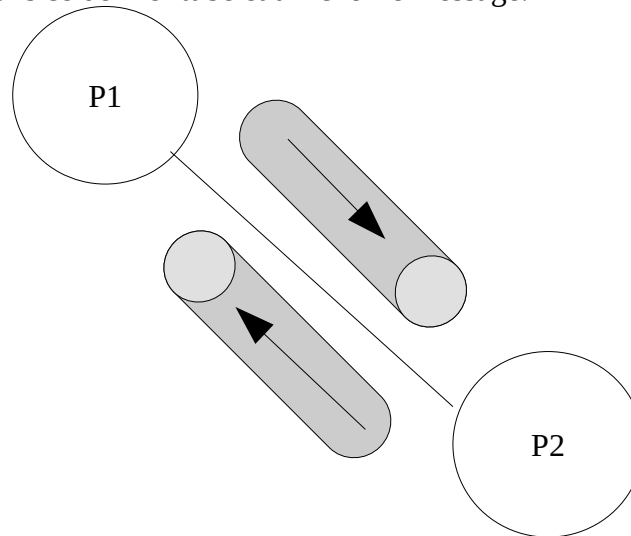


1) En utilisant `popen`, afficher le résultat donné par le programme externe `"/bin/uname"`.

Soit le processus P1 créant un fils P2.

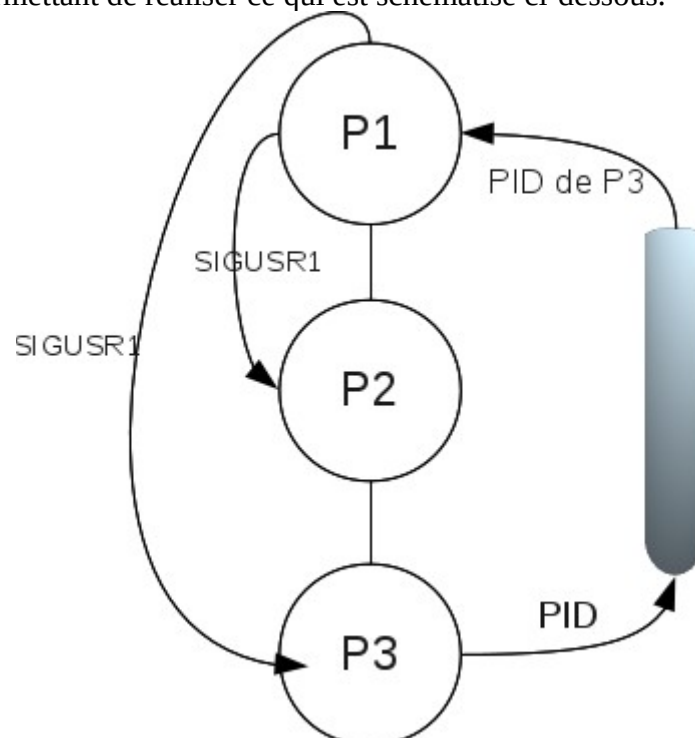
2) Écrivez un programme où P1 écrit un message dans un tube (pipe) à destination de P2.

3) idem, mais en plus, lorsque P2 a lu et affiché le message, il écrit à P1 un autre message dans un autre tube. P1 doit lire dans ce dernier tube et afficher le message.



La problématique d'envoi de signal d'un père vers un petit fils peut être résolue si le petit fils peut communiquer son pid au grand-père.

4) Donnez le code permettant de réaliser ce qui est schématisé ci-dessous.



## READ

```
#include <unistd.h>
```

```
ssize_t read(int fd, void *buf, size_t count);
```

### DESCRIPTION

**read()** lit jusqu'à *count* octets depuis le descripteur de fichier *fd* dans le tampon pointé par *buf*.

Si *count* vaut zéro, **read()** renvoie zéro et n'a pas d'autres effets.

Si *count* est supérieur à SSIZE\_MAX, le résultat est indéfini.

### VALEUR RENVOYÉE

**read()** renvoie -1 s'il échoue, auquel cas *errno* contient le code d'erreur, et la position de la tête de lecture est indéfinie.

Sinon, **read** renvoie le nombre d'octets lus (0 en fin de fichier), et avance la tête de lecture de ce nombre.

Le fait que le nombre renvoyé soit plus petit que le nombre demandé n'est pas une erreur. Ceci se produit à la fin du fichier, ou si on lit depuis un tube ou un terminal, ou encore si **read()** a été interrompu par un signal.

---

## WRITE

```
#include <unistd.h>
```

```
ssize_t write(int fd, const void *buf, size_t count);
```

### DESCRIPTION

**write()** lit au maximum *count* octets dans la zone mémoire pointée par *buf*, et les écrit dans le fichier référencé par le descripteur *fd*.

Le nombre d'octets écrits peut être inférieur à *count* par exemple si la place disponible sur le périphérique est insuffisante, ou la limite de ressource RLIMIT\_FSIZE est atteinte (voir `setrlimit(2)`), ou l'appel est interrompu par un gestionnaire de signal après avoir écrit moins de *count* octets. (Voir aussi `pipe(7)`.)

**POSIX** réclame qu'une lecture avec **read(2)** effectuée après le retour d'une écriture avec **write()**, renvoie les nouvelles données. Notez que tous les systèmes de fichiers ne sont pas compatibles avec POSIX.

### VALEUR RENVOYÉE

**write()** renvoie le nombre d'octets écrits (0 signifiant aucune écriture), ou -1 s'il échoue, auquel cas *errno* contient le code d'erreur.

Si aucune erreur n'est détectée, 0 sera renvoyé sans autre effet de bord. Si *count* est nul et *fd* est un fichier spécial, les résultats sont indéfinis.