

### Héritage d'un tube ordinaire

- ① | Écrire un programme qui crée un tube ordinaire puis un fils. Successivement, le père et le fils s'échangent des messages (*Salut PAPA* et *Salut Fils*) ☐

### Tube sans écrivains

- ② | Écrire un programme qui crée un tube ordinaire `p` et lit dans `p[0]`. Que se passe-t-il si le programme ne ferme pas `p[1]` avant de lire dans `p[0]`? Même question si le programme ferme `p[1]` avant de lire dans `p[0]`? ☐

### Tube sans lecteurs

- ③ | Écrire un programme qui crée un pipe ordinaire `p` et écrit dans `p[1]`. Que se passe-t-il si le programme ne ferme pas `p[0]` avant d'écrire dans `p[1]`? Même question si le programme ferme `p[0]` avant d'écrire dans `p[1]`? ☐

## Tubes nommés

4

Écrire deux programmes qui communiquent par tube nommé.

— Le premier programme effectue les actions suivantes:

1. crée un tube nommé `myTube` en lecture et en écriture pour le user et le groupe
2. demande un descripteur en écriture
3. écrit un message
4. libère le descripteur
5. demande un descripteur en lecture
6. lit un message et l'affiche
7. libère le descripteur
8. supprime la référence `myTube`

— Le deuxième effectue les actions suivantes :

1. demande un descripteur en lecture sur le tube nommé `myTube`
2. lit un message et l'affiche
3. libère le descripteur
4. demande un descripteur en écriture sur le tube nommé `myTube`
5. écrit un message
6. libère le descripteur .

L'ordre de lancement des deux programmes est indifférent. Expliquer pourquoi.

□