

TD2 – Communication inter-processus (IPC)

Objectifs

- Découvrir les différents mécanismes d'IPC.
- Implémenter un échange père ↔ fils avec :
 1. **Pipe**,
 2. **File de messages (System V)**,
 3. **Mémoire partagée + sémaphores**.

1. Rappel théorique

- Les processus ne partagent pas leur mémoire → besoin d'IPC.
- **Pipe** : simple, unidirectionnel.
- **Message Queue** : messages typés, persistants.
- **Mémoire partagée** : rapide, mais nécessite synchronisation (mutex/sémaphores).

2. Exercices

Exercice 1 : Pipe

- Le père envoie une chaîne "Hello fils" au fils via un pipe.
- Le fils l'affiche.
- Observation : que se passe-t-il si on oublie `close(fd[1])` dans le fils ?

Exercice 2 : File de messages

- Créer une file de messages.
- Le père envoie "Salut fils" avec `msgsnd`.
- Le fils reçoit et affiche avec `msgrcv`.
- Nettoyer avec `msgctl(..., IPC_RMID, ...)`.
- Observation avec :

```
ipcs -q
```

Exercice 3 : Mémoire partagée

- Allouer une zone mémoire avec mmap.
- Le père écrit "Bonjour mémoire partagée".
- Le fils lit et affiche.
- Ajouter un **sémaphore** pour garantir l'ordre.

Exercice 4 : Comparaison

- Chronométrer un transfert de **1000 messages** avec un pipe, puis avec une mémoire partagée.
- Question : lequel est plus rapide ? Pourquoi ?

Commandes utiles

- `ipcs -q` : voir les files de messages.
- `ipcrm -q <ID>` : supprimer une file.
- `man 2 pipe`, `man 2 msgsnd`, `man 2 mmap`.