

TP 5

Processus, fork(), wait() et tubes ordinaires

Manipulation 1 (Statut de fin d'un processus fils).

Écrire un programme dans lequel :

- Le processus initial crée un processus fils
- Le processus fils termine à l'aide d'un appel à `exit()`
- À l'aide de `waitpid()` le processus père affiche des informations sur la terminaison du fils.

Tester le succès de vos appels systèmes. Faire des fonctions séparées pour le fils, le père et l'affichage des informations de terminaison.

1. Tester votre programme en passant au `exit` du fils les valeurs 1, 255, 511, 1023, De combien de valeurs différentes dispose-t-on ? Pouvez-vous trouver confirmation de cette réponse dans les pages de man ?
2. Que se passe-t-il lorsque le fils termine à l'aide de `abort()` ?

Manipulation 2 (Communication par tubes ordinaires `pipe()`).

Écrire un programme qui :

- crée un tube à l'aide de l'appel système `pipe()`
- crée un processus fils qui écrit une chaîne de votre choix dans le tube
- lit la chaîne dans le père et l'affiche sur la sortie standard.

Utilisez des fonctions différentes pour les deux types de tâches : écrire ou lire et afficher. Par exemple : `ecrivain(int fd[])`, `lecteur(int fd[])`.

1. Modifier votre programme de manière à ce que deux fils soient créés, qu'ils écrivent chacun une chaîne et que le père réalise en boucle la lecture/affichage.
2. Modifier votre programme de manière à ce que ce soit le père qui écrive une chaîne sur le tube et que les fils réalisent chacun la lecture.
3. Faites trois fils écrivains, trois fils lecteurs (et rien dans le père).