

TP6 Tube - Système d'exploitation

Sereysethy Touch

23 novembre 2017

Nota :

- Que les exercices marqués * sont obligatoires, vous devez rendre vos travaux au plus tard le **30/11/2018**.
- Pour chaque exercice, il faut toujours un fichier Makefile qui permet de compiler vos programmes et éventuellement un fichier `README.md` indiquant comment exécuter votre programme.
- S'il s'agit de documenter un programme, il faut simplement un fichier `README.md`.

Exercice 1*

Écrivez un programme qui crée un tube et puis crée un processus fils.

Le processus père écrit des données "je t'envoie des données" dans un tube et puis affiche son pid et le nombre d'octets écrits. Tant dis que le fils exécute un programme "od" avec l'option "-c" qui récupère les données envoyée par le tube.

Exercice 2*

Écrivez un programme permettant de réaliser ces commandes enchaînées sans utiliser des fonctions `read` ou `write` :

```
ls | sort -d > file.txt
```

Il n'est pas nécessaire de lire ou analyser ces commandes à partir du clavier. Ce n'est pas l'objectif de l'exercice.

Exercices 3* -

Écrivez un programme père qui crée un processus fils. Le père lit des caractères à partir de l'entrée standard, l'envoie au fils et récupère des caractères envoyés par le fils et les affiche sur la sortie standard. Le fils, quant à lui, lit des données transmises par le père, les convertit en majuscule et puis les envoie en retour à son père. Le programme s'arrête seulement quand il n'a plus de données à lire ou à écrire.

Exercice 4* - client/serveur, mini SGBD

Écrivez un programme serveur dico qui permet de stocker des mot et leurs définitions dans un fichier. Le serveur reçoit les commandes de clients via un tube. Voici une liste des commandes que le serveur peut faire :

- ajouter un mot et sa définition
- supprimer un mot
- chercher un mot

A chaque reception de la requête du client, le serveur exécute la commande et puis renvoie le résultat en fonction de la commande. Par exemple, pour la commande "ajouter un mot", le serveur répond simplement si le mot est ajouté avec succès ou avec échec. Le serveur doit aussi indiquer la raison en cas d'échec.

Le serveur peut accepter les requêtes de plusieurs clients.

Si le serveur n'est pas démarré, le client doit terminer l'exécution.

On peut démarrer le serveur avec une option par exemple "-i" pour re-initialiser le fichier de stockage.

Le serveur doit traiter correctement le signal, avant de terminer le programme, le serveur doit prendre soin de sauvegarder tous les mots et libérer toutes les ressources utilisées.

Exercice 5 - optionnel

En reprenant l'exercice `monshell`, on veut ajouter des pipes entre commandes et la redirection de sortie d'une commande vers un fichier ou la redirection de l'entrée d'une commande à partir d'un fichier.

Mon shell accepte donc ce genre de commandes :

```
monshell$ cmd arg1 arg2
monshell$ cmd arg1 arg2 < entree > sortie
monshell$ cmd1 arg1 arg2... | cmd2 args...
```