

IN405 – Système d’exploitation

TD 7 – Tubes

S. Gougeaud

2017/2018

Exercice 1 – Premiers pas avec les tubes

L’objectif de l’exercice est d’apprendre à utiliser les tubes anonymes et nommés à travers le transfert de simples données entre deux processus. Pour les énoncés suivants, implémentez une version en utilisant les tubes anonymes, et une seconde avec les tubes nommés :

1. Transfert d’une chaîne de caractères de taille 16 max (exemple : "Hello World!").
2. Transfert d’une paire d’entiers.

Exercice 2 – Mécanisme de barrière

L’objectif de l’exercice est d’offrir à l’utilisateur un mécanisme de barrière entre n processus issus d’un même processus père, à l’aide de tubes (les mutex et conditions n’étant pas disponibles en mémoire non partagée). Chaque processus fils attend un nombre aléatoire de secondes compris entre 1 et 10, puis entre dans la barrière. Le père agit comme coordinateur, attendant que tous les fils soient dans la barrière pour les libérer.

Exercice 3 – Affichage informatisé

L’objectif de l’exercice est de faire passer un message à un processus indépendant (ne faisant pas partie de la même hiérarchie de processus), pour qu’il puisse l’afficher sur le terminal. L’exercice se découpe en plusieurs parties :

1. Programme 'affichage' – Ce programme doit afficher sur le terminal, toutes les 2 secondes, l’un des messages qu’il possède dans sa base de données (tableau de chaînes de caractère). Les messages sont lus les uns après les autres, et la lecture reprend au début de la base si on vient de lire la dernière entrée. Initialement, la base est constituée de deux messages :
 - "Bonjour, ceci est un affichage informatisé" ;

- "dd/MM/yy – hh:mm:ss" pour le jour et l'heure (`man gettimeofday` et `man localtime`).
2. Réception d'un signal de 'affichage' – A la réception du signal, le programme 'affichage' peut, en fonction de sa nature, soit ajouter à sa base la nouvelle information présente dans le tube, soit remettre la base dans sa configuration initiale ie. constitué des deux messages décrits précédemment. Pour ceci, vous pouvez utiliser les signaux `SIGUSR1` et `SIGUSR2`.
 3. Programme 'envoi' – Ce programme doit envoyer au programme affichage des signaux en fonction de ses arguments. Il possède trois arguments : l'action à réaliser (remise à zéro ou ajout de message), le PID du programme affichage (pour ceci, l'afficher en début d'exécution) et le message à ajouter si besoin.
 4. Quelques conseils :
 - La réception d'un signal entraînant une interruption de l'exécution normale du processus, la fonction 'handler' ne peut accéder qu'aux variables globales du programme : stockez le tableau (et autre(s) variable(s) NECESSAIRE(S)) dans des variables globales.
 - Les fonctions `open()` sur des tubes nommés sont bloquantes tant qu'il n'est pas ouvert dans les deux directions (lecture et écriture), n'ouvrez le tube dans le programme 'affichage' que si c'est nécessaire.
 - Le programme 'affichage' contient une boucle affichant périodiquement les messages de son tableau. Vous pouvez choisir de faire une boucle infinie ou une boucle se répétant un nombre défini de fois (20, 50, 100, etc.). Dans tous les cas, le raccourci `CTRL+c` envoie un signal de terminaison au processus s'exécutant au premier plan dans le terminal.
 5. Pour aller plus loin, faites que le programme 'envoi' puisse terminer l'exécution du programme 'affichage' à l'aide d'un signal.