

SR2 – Partie « Systèmes » - Préparation Séance 2 de TP Processus Unix & Tubes

Exercice 1 – Révisions Tubes – A faire avant la séance 2

Écrire un programme dans lequel un processus père envoie à son fils un message un certain nombre de fois NE via **un tube** de communication. Le fils ne connaît pas NE et doit se terminer quand il ne pourra plus recevoir de messages. Le père doit se terminer en **dernier**.

Le nombre NE d'envois à réaliser constitue le **paramètre** de cette application.

Attention : Réfléchir au **schéma** de communication avant de coder.

Exemple d'exécution : ./exo1 3

/ Le message est envoyé 3 fois au fils */*

```
Pere - Message 1 envoye
      Fils - Recu de mon pere : Je suis ton pere...
Pere - Message 2 envoye
      Fils - Recu de mon pere : Je suis ton pere...
Pere - Message 3 envoye
      Fils - Recu de mon pere : Je suis ton pere...
      Fils - Je me termine
Pere - Je me termine en dernier
```

Exercice 2 – Révisions Tubes – A faire avant la séance 2

Écrire un programme dans lequel NF fils jouent le rôle de capteurs de véhicules et envoient un message à leur père, à **chaque fois** que NBV véhicules ont été détectés. Lorsque NBVM véhicules ont été comptés, les fils se **terminent**.

Le père joue le rôle d'un panneau d'affichage en affichant les informations **au fur et à mesure** qu'elles lui arrivent. Il se **termine** quand ses fils se sont terminés (en d'autres termes, il n'a pas à connaître NBVM).

Le nombre NF de capteurs (fils), le nombre maximum de véhicules NBVM et le nombre NBV déclenchant l'envoi constituent les **paramètres** de cette application.

Le passage d'un véhicule sera symbolisé par la saisie d'un caractère au clavier.

Attention : Réfléchir au **schéma** de communication avant de coder.

```
ab
Pere (96) - Capteur 1 : nombre de vehicules = 1
      Capteur 1 (98) : 1 vehicules de plus => 1
      Capteur 0 (97) : 1 vehicules de plus => 1
Pere (96) - Capteur 0 : nombre de vehicules = 1
      Capteur 1 (98) : 1 vehicules de plus => 2
Pere (96) - Capteur 1 : nombre de vehicules = 2
cde
      Capteur 0 (97) : 1 vehicules de plus => 2
      Capteur 1 (98) : 1 vehicules de plus => 3
Pere (96) - Capteur 1 : nombre de vehicules = 3
      Capteur 0 (97) : 1 vehicules de plus => 3
      Capteur 1 (98) : 1 vehicules de plus => 4
Pere (96) - Capteur 0 : nombre de vehicules = 2
      Capteur 1 (98) : Termine
Pere (96) - Capteur 0 : nombre de vehicules = 3
Pere (96) - Capteur 1 : nombre de vehicules = 4
fg
      Capteur 0 (97) : 1 vehicules de plus => 4
Pere (96) - Capteur 0 : nombre de vehicules = 4
      Capteur 0 (97) : Termine
Pere (96) - Je me termine en dernier
```

Exemple d'exécution : ./exo2 2 4 1

/ 2 capteurs, envoyant leur valeur tous les 1 véhicule, à concurrence de 4 captés.*

*On a tapé au clavier les caractères ab<entrée>cde<entrée>fg<entrée> afin de simuler le passage des véhicules */*