UNIVERSITÉ
DE TOULOUSE

Licence Informatique

Université Paul Sabatier
118 route de Narbonne
31062 Toulouse Cedex 09

SR2 – Partie « Systèmes » - Travaux pratiques n°0

Révisions – A faire avant la première séance de TP

Processus Unix – Tubes de communication

Exercice 1 – main() paramétré & Processus Unix (sans les tubes)

Écrire une application dans laquelle NF processus fils sont créés par le processus principal (père).

Chaque processus affiche un message NM fois, en s'identifiant, avant de se terminer en retournant à son père son rang de création.

Le processus père devra afficher les informations retournées au fur et à mesure que les processus fils se terminent, accompagnée de leur identité.

L'application sera paramétrée par les valeurs de NF et de NM. Le fait que les paramètres sont bien fournis lors de l'appel doit (toujours) être vérifié.

Exemples d'exécution : ./exo1 2 4

/* 2 fils affichent 4 fois un message */

```
Activite rang 0 : identifiant = 139
Activite rang 0 : identifiant = 139
Activite rang 1 : identifiant = 140
Activite rang 0 : identifiant = 139
Activite rang 1 : identifiant = 140
Activite rang 0 : identifiant = 139
Activite rang 1 : identifiant = 140
Valeur retournee par le fils 139 = 0
Activite rang 1 : identifiant = 140
Valeur retournee par le fils 140 = 1
```

Exercice 2 – Processus & Tubes : Emballage

On considère une application dans laquelle un processus père et NF fils s'exécutent en parallèle.

Le père est une « usine » qui produit NB fois, toutes les NS secondes, une certaine quantité de chocolats qu'il met à disposition de ses NF fils qui les emballent.

Chaque fils produit une boîte de chocolats de taille proportionnelle à son rang de création (le fils 1 produit des boîtes de 6 chocolats, le fils 2 de 12, etc.). À chaque boîte complétée, il affiche un message indiquant la fabrication d'une boîte. Avant de se terminer, il affiche le nombre total de boîtes qu'il a produites.

L'application doit être paramétrée par les valeurs de NF, NB et de NS, et se **terminer** d'elle-même.

Remarque : le père pourra utiliser la fonction *unsigned int sleep(unsigned int sleep);* pour s'endormir pour une durée déterminée (les NS secondes ici).

Étape 1 : Faire une **description schématique** de la solution proposée pour faire communiquer les processus.

Étape 2 : Écrire le code.

Exemples d'exécution : ./emballage 4 3 1

/* 4 fils emballent, le père produit 3 fois, toutes les 1 secondes */

```
Usine : je produis 60 chocolats [Wed Mar 19 17:16:34 2025]
Emballage 0 : nouvelle boite de 6 produite
Emballage 0 : nouvelle boite de 6 produite
Emballage 0 : nouvelle boite de 6 produite
Emballage 0 : nouvelle boite de 6 produite
Emballage 0 : nouvelle boite de 6 produite
Emballage 0 : nouvelle boite de 6 produite
Emballage 1 : nouvelle boite de 12 produite
Usine : je produis 60 chocolats [Wed Mar 19 17:16:35 2025]
Emballage 0 : nouvelle boite de 6 produite
Emballage 2 : nouvelle boite de 18 produite
Emballage 0 : nouvelle boite de 6 produite
Emballage 2 : nouvelle boite de 18 produite
Emballage 0 : nouvelle boite de 6 produite
Emballage 0 : nouvelle boite de 6 produite
Usine : je produis 60 chocolats [Wed Mar 19 17:16:36 2025]
Emballage 2 : nouvelle boite de 18 produite
Emballage 2 : nouvelle boite de 18 produite
Emballage 2 : nouvelle boite de 18 produite
Usine : je produis 60 chocolats [Wed Mar 19 17:16:37 2025]
Emballage 2 : nouvelle boite de 18 produite
Emballage 2 : nouvelle boite de 18 produite
Emballage 3 : nouvelle boite de 24 produite
Emballage 2 : j'ai produit 7 boites de 18 chocolats
Emballage 0 : j'ai produit 10 boites de 6 chocolats
Emballage 3 : j'ai produit 1 boites de 24 chocolats
Emballage 1 : j'ai produit 1 boites de 12 chocolats
```

Remarque : Selon les entrelacements, il se peut que les fils ne puissent pas compléter leur boîte, il est probable que lors de certaines exécutions, tous les chocolats produits ne soient pas « emballés ».

Exercice 3 – Processus & Tubes : Appel d'offre

On considère une application dans laquelle un processus père et NF fils s'exécutent en parallèle.

Le père est un « client » qui souhaite établir un devis pour l'achat d'une certaine quantité d'un produit donné.

Les fils représentent les « fournisseurs » capables de vendre ce produit et d'établir le devis demandé.

Écrire une application dans laquelle le client envoie aux fournisseurs potentiels une (unique) demande de devis comportant les informations suivantes : référence d'un produit (2 lettres) et quantité demandée.

Chaque fournisseur répondra au client avec un devis (unique) rappelant la référence et la quantité demandés et indiquant le montant total de la commande (un réel). Pour simplifier, on supposera que ce montant ne peut être supérieur à MONTANT_MAX.

Après avoir étudié toutes les réponses, le client devra afficher le montant du fournisseur le mieux offrant (prix le plus bas). Il devra aussi avertir ce dernier qu'il a remporté le marché.

Le fournisseur ayant remporté le marché affichera cette information avant de se terminer. Les autres afficheront qu'ils feront mieux la prochaine fois.
L'application doit comporter en paramètre la valeur de NF et se **terminer** d'elle-même.

Remarque : On pourra utiliser les fonctions *void srand(unsigned int seed);* et *int rand(void);* pour générer un entier aléatoire lors de la définition du prix de vente d'un produit (voir man). La graine sera le pid du processus appelant *srand().*

Étape 1 : Faire une **description schématique** de la solution proposée pour faire communiquer les processus.
Étape 2 : Écrire le code.

Exemples d'exécution : ./appel 2
/* 2 fils « fournisseurs » */

```
Client : J'envoie demande au fournisseur 0
Fournisseur 0 : j'ai recu la demande (AB, 5)
Client : J'envoie demande au fournisseur 1
Fournisseur 0 : montant = 2.500000
Fournisseur 1 : j'ai recu la demande (AB, 5)
Client : Je recois 2.500000 du fournisseur 0
Fournisseur 1 : montant = 20.000000
Client : Je recois 20.000000 du fournisseur 1
Client : Je choisis le fournisseur 0 pour un montant de 2.500000
Fournisseur 1 : j'ai perdu le marche avec 20.000000
Fournisseur 0 : j'ai obtenu le marche pour 2.500000
```