

Université de Bretagne Sud
UFR SSI, Dept MIS
SYSTEME D'EXPLOITATION
CONCURRENCE
Les processus Unix
Les appels système fork, wait, exit et exec...

Luc Courtrai

Email : Luc.Courtrai@univ-ubs.fr

1 Multif

Ecrire un programme C qui affiche en parallèle sur la sortie standard le contenu des fichiers dont les noms sont passés en arguments de la commande. Chaque fichier sera pris en charge par un processus distinct.

2 Serveur de jobs

On désire faire exécuter à un programme des tâches en parallèle. Pour simplifier l'exercice chaque tâche aura à imprimer un caractère sur la sortie standard à un intervalle de temps. Chaque tâche doit être prise en charge par un processus.

La liste des tâches est décrite dans un fichier d'ordres dont voici un exemple (*donnees.jobs*) :

```
a 10 2  
b 20 1  
c 15 3
```

Dans cet exemple, la première tâche va consister à imprimer 10 caractères *a* sur la sortie standard. Entre chaque impression de caractère, au moins 2 secondes doivent s'écouler.

Le programme *pr* lit les données sur l'entrée standard :

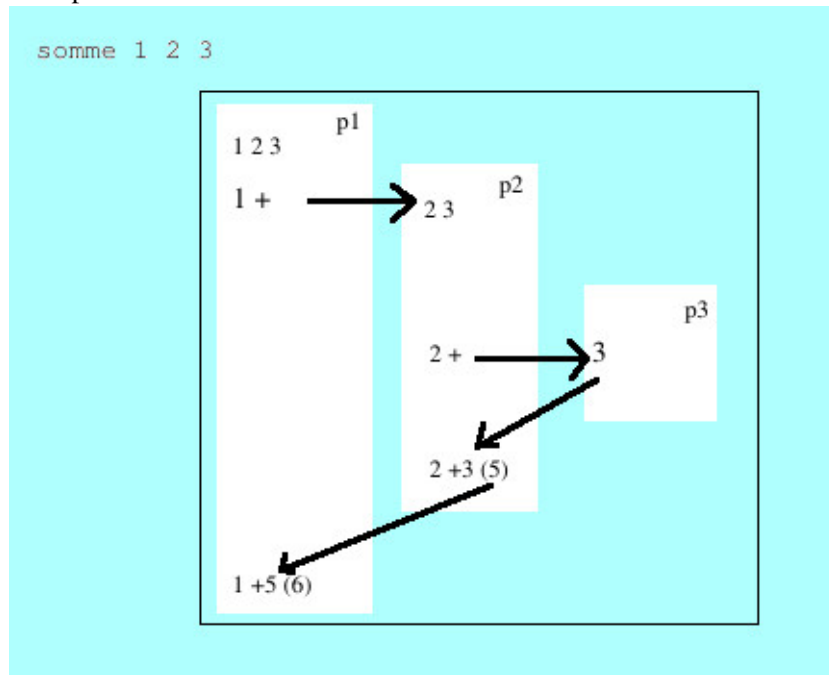
```
pr < donnee.jobs
```

Ecrire en C le programme *pr*.

3 Répartition d'un calcul

On désire répartir le calcul d'une addition d'une liste d'entiers sur plusieurs processus. Chaque processus effectue la somme du premier de la liste avec le reste de la liste ; la somme du reste (sous liste) étant calculée par d'autres processus de la même façon. Chaque processus ne fait qu'une addition.

exemple : somme 1 2 3



La transmission de la réponse se fera grâce au code de sortie (*exit*) et le (*wait*).