# Travaux Pratiques de Calcul Distribué n°2

**Cours d'Informatique de l'École Centrale d'Électronique**

*—Ingénieur Quatrième Année—*

---

L'objet de cette feuille est de mettre en place un programme distribué. Plus précisément, ce TP porte sur la simulation du problème de Josephus à l'aide d'un anneau de processus. L'objectif de cette simulation est de découvrir quelques outils disponibles permettant de visualiser ou analyser une exécution en environnement `PVM`.

---

▸ **Exercice 1.** (problème de Josephus) *Le problème est le suivant : on dispose les entiers de* 1 *à* $n$ *autour d'un cercle et on se donne un entier* $k$*. On élimine alors les entiers successivement en commençant par le numéro* $k$*. La procédure d'élimination consiste ainsi à tourner autour du cercle et à éliminer un élément sur* $k$ *jusqu'à ce qu'il ne reste plus qu'un seul entier. On appelle alors* $J(n,k)$ *le dernier entier restant sur le cercle. Par exemple, pour* $n = 10$ *et* $k = 2$*, on élimine au premier tour les entiers* $2, 4, 6, 8, 10$ *de sorte qu'il reste* $1, 3, 5, 7, 9$*. Ensuite, on élimine* $3, 7$ *puis* 1 *et enfin au troisième tour c'est* 9 *qui est éliminé. On a donc* $J(10, 2) = 5$ *car* 5 *est le dernier entier restant autour du cercle.*

*On souhaite simuler cet algorithme à l'aide de* `PVM` *de telle sorte que chaque entier corresponde à un processus. Les processus vont être organisés en un anneau et chaque processus ne va connaître que le numéro (c'est-à-dire essentiellement le* `tid` *correspondant) de son successeur. Un processus maître va se charger de gérer le calcul.*

1. *Créer un programme* `josephus_master` *prenant en argument sur sa ligne de commande les deux paramètres entiers* $n$ *et* $k$ *et calculant* $J(n,k)$*. Le programme lancera un programme* `josephus_slave` *qui créera* $n-1$ *autre processus* `josephus_slave` *formant un anneau, chaque processus devant connaître le* `tid` *de la tâche* `josephus_master` *et le* `tid` *de son successeur dans l'anneau. Une fois l'anneau créé, le programme* `josephus_master` *initiera le calcul de* $J(n,k)$ *en envoyant un message au premier* `josephus_slave` *puis il se mettra à l'écoute d'un message lui signifiant le numéro du dernier processus.*

2. *Créer le programme* `josephus_slave` *simulant le rôle d'un entier. Ce programme aura tout d'abord un comportement visant à créer l'anneau jusqu'à ce qu'il soit complet puis il rentrera dans une boucle d'attente d'événements. On pourra prévoir deux types d'événements, le premier concernant la terminaison d'un processus et le deuxième une demande du* `tid` *du successeur afin de maintenir la structure d'anneau lors de la suppression d'un processus. Le processus communiquera son numéro au* `josephus_master` *s'il est devenu son propre successeur ou en d'autres termes s'il est le dernier autour du cercle.*

3. *Visualiser avec* `XPVM` *et* `PGPVM2` *le comportement du programme. Quelles sont les fenêtres intéressantes pour faire une analyse des performances ?*

4. *Ajouter des labels aux deux programmes* `josephus_master` *et* `josephus_slave` *afin de tracer certaines parties intéressantes de ces deux processus.*