

Systeme M1 — TP 8 : Drapeau hollandais

Semaine du 16 novembre 2009

Présentation Le problème du drapeau hollandais (Dijkstra) peut être formulé ainsi. On considère n paquets de p jetons tels que tous les jetons d'un même paquet soient de la même couleur, les couleurs associées à chaque paquet étant différentes. Les jetons sont mélangés et repartagés en n paquets de p jetons, sans condition de couleur. On désire reconstituer les n paquets d'origine en procédant de la façon suivante :

- n processus doivent chacun reconstituer un paquet d'une couleur donnée ;
- ces n processus sont organisés en anneau de telle sorte que chacun a un voisin de gauche et un voisin de droite avec lesquels il communique par tubes ;
- chacun reçoit de son voisin de droite les jetons dont celui-ci veut se débarrasser car ils ne sont pas de la bonne couleur ;
- la transmission de jetons est rythmée par une horloge externe : à chaque impulsion d'horloge, chaque processus de l'anneau transmet à son voisin de gauche un jeton dont il ne veut pas (s'il en possède).

En particulier étant donné des jetons rouges, blancs et bleus on peut vouloir les regrouper de sorte que les jetons de la même couleur soient adjacents et que les couleurs soient dans l'ordre du drapeau hollandais.

Exemple Si on représente les couleurs par des entiers, et des paquets par des tableaux de taille n dont la somme des éléments est p alors $2\ 0\ 1$ représente un paquet de deux objets de couleur 0 et un objet de couleur 2.

Les paquets des n processus peuvent être vus comme une matrice carrée de taille n , où $\text{jetons}[i][j]$ est le nombre de jetons de couleur j chez le processus i .

L'évolution d'une configuration ($n=3, p=4$) à chaque tick d'horloge :

Debut	Processus 0	2 0 1	2 1 0	3 0 0	3 1 0	3 0 0	Fin
	Processus 1	1 1 1	0 1 2	0 2 1	0 2 0	0 3 0	
	Processus 2	0 2 1	1 1 1	0 1 2	0 0 3	0 0 3	

Pour commencer, le processus 0 qui veut reconstituer les jetons de couleur 0, se débarrasse du jeton de couleur 2 – et ensuite reçoit un jeton de couleur 1 du processus 2, etc.

Exercice On se propose de décomposer la solution en quatre modules différents.

- le programme `hollandais` est le code du processus principal, qui initie la création de l'anneau de processus en créant un fils puis se recouvre par le programme `horloge` ;
- le programme `horloge` envoie un top toutes les secondes à tous les processus de l'anneau, jusqu'à ce que la terminaison soit détectée ;
- la fonction `creer_anneau(char *nbprocs, char *nbjetons)`, est exécutée par le fils du processus principal pour créer l'anneau de `n` processus reliés par tubes, avant que chacun de ces processus ne se recouvre par `tache` ;
- le programme `tache`, exécuté par chaque processus de l'anneau après le recouvrement et rythmé par les impulsions de l'horloge, réalise les échanges de jetons jusqu'à ce que les paquets soient reconstitués.

1. Consulter le fichier `hollandais.c` qui est également fourni. Vous constaterez en particulier qu'il crée le premier processus de l'anneau et que celui-ci devient leader d'une nouvelle session. Pourquoi ? Comment tester si une nouvelle sessions a été créée ? Pour quelle raison les signaux `SIGUSR1` et `SIGCHLD` sont-ils masqués ?
2. Consulter le code de l'horloge. Noter qu'elle prend en argument le numéro du groupe de processus avec lequel elle doit interagir. Exécutez la. Quel signal attend-elle ? Envoyer ce signal en utilisant la commande shell `kill`. A quoi sert la constante `TEMPO` ?
3. Consulter le fichier `creer_anneau.c`. La fonction `creer_anneau` prend deux paramètres de type `char *` définissant le nombre de processus de l'anneau et le nombre de jetons de chaque paquet. Noter que les lectures dans les tubes doivent être non-bloquantes, pour éviter le blocage de processus n'ayant reçu aucun jeton (voir man `fcntl`). La fonction `creer_anneau` lit la répartition initiale des jetons dans le fichier `jetons` qui utilise le format du petit exemple du début.
4. Écrire le code de `tache`, qui prend 4 arguments : le numéro (logique) du processus dans l'anneau (0 pour le leader, 1 pour son premier fils...), le nombre de processus de l'anneau, le nombre de jetons par paquet, et la constitution du paquet initial (au format du fichier exemple `jetons`).

On prendra soin de bien synchroniser les processus mis en jeu entre les différentes étapes du programme. On devra en particulier être capable de détecter la fin de l'échange des jetons.