

Université de Rouen

I.U.P. 2

Module : Système I

## T.P. 10 : Tubes et tubes nommées

**Exercice 1** Le but de cet exercice est d'écrire un client et un serveur communiquant par tubes nommées. Le client et le serveur devront se synchroniser sur l'ouverture des tubes. Le serveur devra fournir par l'intermédiaire d'un tube le fichier que le client lui demandera. Cet envoi se fera par blocs. Le client précisera le nom du fichier qu'il veut obtenir, le serveur commencera par lui envoyer la taille de celui-ci puis son contenu. Le client écrira le fichier reçu sur le disque et se terminera en fin de transfert. Le serveur se remettra en attente d'un nouveau client.

**Exercice 2** On définit  $n$  tâches ( $0..n - 1$ ) différentes, chaque tâche consiste en la reconstitution d'un paquet d'une couleur donnée.

- Les tâches sont organisées circulairement de telle sorte qu'une tâche  $i$  a une voisine de gauche  $i - 1$  (si  $i \neq 0$ , sinon  $n - 1$ ) et une voisine de droite  $i + 1$  (si  $i \neq n - 1$ , sinon  $0$ ). Une tâche reçoit de sa voisine de droite les jetons dont elle veut se débarrasser car ils ne sont pas de la bonne couleur; de son côté elle envoie à sa voisine de gauche les jetons dont elle veut se débarrasser. On transmet les jetons entre les tâches via des tubes.
- Pour que le système s'arrête, l'une des tâches est privilégiée en ce sens qu'elle dispose au départ de deux jetons spéciaux  $F$  (fin) et  $S$  (stop) qui seront utilisés de la manière suivante :
  - le jeton  $F$  est mis en circulation par la tâche privilégiée lorsque celle-ci a rempli son contrat; la tâche qui détient ce jeton à un instant donné le fera à son tour circuler lorsqu'elle aura rempli son contrat.
  - Lorsque la tâche privilégiée au départ récupère le jeton  $F$ , c'est-à-dire que toutes les tâches ont terminé leur travail, elle met en circulation le jeton  $S$  qui a pour rôle de stopper les autres tâches.

Ecrire l'ensemble des fonctions permettant de réaliser cette stratégie.

*tache\_privilegiee()* { ..... }

*tache\_autre()* {..... }

*main()* {..... }