

Programmation Concurrente

La banque de Grütschli

Travail Pratique

But

Réaliser la modélisation d'un problème concurrent à l'aide de sémaphores.

Enoncé du problème¹

Au fin fond d'une vallée grisonnaise perdue, se trouve, Grütschli, un petit village isolé avec une population constante N . Dans ce village, une banque offre des intérêts composés toutes les heures. Les habitants ont donc pris l'habitude de porter peu de monnaie avec eux et d'effectuer continuellement des transactions bancaires durant la journée.

La banque utilise un système de tickets semblable à la Poste : en entrant dans la banque le client prend un ticket lui donnant son ordre de passage, et dès que le guichet se libère, le guichetier appelle le prochain client. Les clients sont ainsi servis selon une politique premier arrivé, premier servi (FIFO).

Certains clients sont malins et économes de leur temps ; ils prennent un numéro, mais sortent de la banque pour vaquer à d'autres occupations avant de revenir et se réinsérer dans la file pour se faire servir. Toutefois si leur numéro a déjà été tiré, ils en reprennent un nouveau et recommencent. Les clients se comportent donc de la manière suivante :

```
1      boucle infinie {
2      ———faire une activité d'une durée  $d_0$ 
3      ———entrer dans la banque
4      ———prendre un ticket  $t$ 
5      ———si  $t - t_{servi} > 3$  alors {
6      —————tirer un nombre aléatoire  $n$  entre 0 et 1
7      —————si  $n < p$  alors {
8      —————sortir de la banque
9      —————faire une activité d'une durée  $d_1 * (t - t_{servi})$ 
10     —————revenir à la banque
11     —————si  $t < t_{servi}$  alors {
12     —————aller en 4
13     —————}
14     —————}
15     ———}
16     ———attendre son tour (càd attendre de se faire réveiller par le guichetier)
17     ———se faire servir
18     }
```

Ainsi, un client se présentant à la banque prend son ticket t (ligne 4), et s'il y a plus de 3 clients avant lui (y compris celui en cours de service), il peut décider de faire une autre activité au lieu

¹ Ce TP est inspiré d'une proposition de Yann Thoma

d'attendre inutilement. Ce choix se fait par un tirage aléatoire avec une probabilité de réussite égale à p (ligne 7). Cette probabilité est une constante. Le guichetier se comporte plus simplement :

```
1   boucle infinie {
2   ————  $t_{servi} = t_{servi} + 1$ 
3   ———— si client  $t_{servi}$  est présent alors {
4   ———— ———— appeler le client et le servir pendant une durée  $d_1$ 
5   ———— } sinon {
6   ———— ———— si la salle d'attente est vide  $\rightarrow$  attendre qu'un client entre dans la banque
7   ———— ———— sinon  $\rightarrow$  servir le prochain client
8   ———— }
9   }
```

Cahier des charges

Réalisez le programme énoncé avec les contraintes ci-dessous.

- Les durées d_0 et d_1 sont des attentes passives (fonction `sleep`), avec la relation $d_0 = 10 * d_1$. Pour tester votre modélisation, il sera peut-être utile d'utiliser des variables aléatoires.
- La banque ne comporte qu'un seul guichet et les clients ont tous le même comportement.
- La population du village N , la valeurs d'attente d_1 ainsi que la probabilité p seront toutes trois passées en argument à votre programme via la ligne de commande. Veillez à indiquer la syntaxe de votre programme à l'utilisateur si nécessaire.
- Il n'est pas demandé de prévoir une terminaison à votre programme, et on supposera qu'il n'y a pas de débordement de tickets. Autrement dit, le nombre de clients entrant dans la banque durant une exécution ne dépassera pas $2^{32}-1$.
- Quand un client se fait servir, il est synchrone avec le guichetier. Autrement dit, le client est en attente passive pendant toute la durée de son service et c'est le guichetier qui le réveille.
- Le guichetier ne devra jamais attendre l'arrivée d'un client alors qu'il y en a en attente.
- Votre modélisation ne devra comporter aucune attente active.
- Il n'est pas demandé de générer des sorties textuelles décrivant les événements d'exécution du programme.
- Il est toutefois suggéré que vous le fassiez pour vos propres tests potentiellement utiles au développement de l'application.

Travail à rendre

Les modalités de rendu sont explicitées dans le document « `Consignes Travaux Pratiques.pdf` » disponible sur CyberLearn.

Toutefois, pour ce TP vous ne rendrez **aucun rapport**.