

INF8541 : Devoir #8

(À remettre le 27 novembre)

1. Modules pour un CompteBancaire

Un (unique) compte bancaire est *partagé* par un groupe de personnes. Chaque personne peut déposer ou retirer des fonds de ce compte. Le solde courant à un instant donné est la somme de tous les dépôts ayant été effectués à date moins la somme de tous les retraits qui ont été acceptés. Ce solde ne peut et ne doit jamais être négatif (pas de marge de crédit). Une opération de retrait doit être *retardée* (bloquée) si le montant à retirer est supérieur au solde courant.

On veut définir un module logiciel, dans la notation Andrews, pour réaliser un tel compte bancaire. Les en-têtes des deux opérations exportées par le module seront simplement les suivantes :

```
deposer( float montant );  
retirer( float montant );
```

Dans les deux cas, il s'agit donc de procédures qui ne retournent *aucun* résultat. De plus, pour simplifier, il n'y a pas d'opérations pour examiner le solde du compte et un montant est simplement un nombre point-flottant.

Vous devez définir différentes versions de ce module. Plus précisément, vous devez définir deux (2) mises en oeuvre parmi les trois (3) possibilités suivantes :

1. Un processus de gestion du compte bancaire avec lequel on ne peut communiquer que par l'intermédiaire de canaux de communications (type `chan`) (chap. 7).
2. Un module utilisable par l'intermédiaire de RPC et utilisant des sémaphores pour assurer l'exclusion mutuelle et les synchronisations conditionnelles (section 8.1)
3. Un module utilisant l'approche par rendez-vous (section 8.2).

Dans chaque cas, il s'agit simplement de définir un module qui gère *un unique* compte bancaire. Évidemment, dans chaque cas, vous devez aussi vous assurer que le comportement est correct en présence de requêtes multiples faites possiblement de façon concurrente.

Note importante : Si plusieurs retraits sont en attente et qu'un dépôt est effectué, vous pouvez utiliser *n'importe quelle stratégie* pour sélectionner le ou les retraits qui seront satisfaits; vous n'avez donc pas à respecter l'ordre d'arrivée des retraits ni à tenter de satisfaire le plus grand nombre de retraits.