

LOG8470

Méthodes formelles en fiabilité et en sécurité

TP 1 : Synchronisation des systèmes concurrents

Session	Automne 2021
Pondération	5% de la note finale
Taille des équipes	2 étudiants
Date de remise	21 septembre (23h55 au plus tard)
Directives particulières	Soumission des fichiers promelas par Moodle uniquement dans un dossier zip sous la forme tp1-matricule1_matricule2.zip . Toute soumission du livrable en retard est pénalisée à raison de 10% par jour de retard.
Moyen de communication	<ul style="list-style-type: none">- Serveur discord- Courriel : basil.capar@polymtl.ca

1. Notions du cours à connaître

- Communication par variables partagées
- Synchronisation par canal

2. Objectif

Ce laboratoire est séparé en 2 exercices. L'objectif du premier est d'implanter un système qui permet de gérer des variables partagées sans qu'il y ait de *deadlock*. Le second exercice a pour but de modéliser un système de synchronisation de micro services.

3. Communication par variable partagée : Dîner entre amis (2 pts)

Le problème du dîner entre amis est inspiré du très connu dîner des philosophes. En effet, il y a 2 amis qui ont une assiette de lasagne devant eux. Chaque personne a une fourchette. Cependant, il y a un seul couteau pour les 2 personnes. Ainsi, ils devront se partager le couteau. Il faut savoir qu'on doit éviter qu'une personne meure de faim. Le cas échéant, il y a famine. Votre objectif est de concevoir un programme en Promela qui s'assure qu'il n'y a pas d'exclusion mutuelle.

4. Synchronisation par canal : Système de commande en ligne (3 pts)

Maintenant que la majeure partie des cours sont en présentiel, il y aura de plus en plus d'étudiants qui seront tentés d'acheter de la nourriture à la cafétéria de l'école. À cet effet, l'école

a décidé de concevoir une application qui permettrait de consulter, commander et livrer des plats de la cafétéria en ligne.

Votre objectif est de concevoir une application en micro services avec Promela. Les 4 services et le dispatcher à implanter sont : Dispatcher (init), MenuService, OrderFoodService, SchoolCreditCardAccountService et DeliveryService.

1. Dispatcher

C'est la fonction init du langage Promela. Cette fonction consiste essentiellement à :

- Initialiser les canaux de communication
- Initialiser les services
- Appeler les services via les canaux
- Écouter les messages de retour via les canaux

En effet, il est possible d'appeler les services autant de fois que désiré. Ainsi, il est logique de dire que le *dispatcher* doit écouter continuellement les réponses des services.

Le *dispatcher* peut **envoyer** les messages suivants :

- AskMenu
- OrderFood
- PayFood
- AskDelivery

Le *dispatcher* **reçoit** les messages suivants :

- CheckedMenu
- PreparingOrder
- FoodPaid
- ChooseDeliveryOrder

2. MenuService

Ce service permet à l'utilisateur de simplement consulter le menu du jour. Il n'est pas demandé de créer un menu. En effet, un **simple affichage dans le terminal que le menu est en consultation** est suffisant pour le cadre du laboratoire.

Le service MenuService peut **envoyer** les messages suivants :

- CheckedMenu

Le service MenuService **reçoit** les messages suivants :

- AskMenu

3. OrderFoodService

Ce service permet à l'utilisateur de commander un plat. Après avoir commandé son plat, l'utilisateur est obligé de payer son plat. Il faut savoir qu'il y a 2 façons de commander un plat :

- L'utilisateur peut commander en ligne et aller chercher son plat dans la cafétéria.
- L'utilisateur peut commander en ligne et se faire livrer son plat à l'endroit où l'utilisateur se situe.

Encore une fois, un simple affichage dans la console pour indiquer que l'utilisateur est entrain de commander est suffisant.

Le service OrderFoodService peut **envoyer** les messages suivants :

- PreparingOrder

Le service OrderFoodService **reçoit** les messages suivants :

- OrderFood

4. SchoolCreditCardAccountService

Ce service permet à l'utilisateur de payer sa commande. Il faut savoir qu'il y a 2 types de paiements possibles :

- L'utilisateur peut payer pour un plat qui ne sera pas livré
- L'utilisateur peut payer pour un plat qui sera livré.

Dans les deux cas, l'utilisateur devra utiliser la carte de crédit de l'école pour payer son plat en ligne avant de chercher/recevoir son plat. Il faut savoir qu'un utilisateur peut effectuer un paiement avec la carte de crédit sans avoir commandé en ligne. En effet, cela signifie que l'utilisateur a commandé en personne.

Le service SchoolCreditCardAccountService peut **envoyer** les messages suivants :

- FoodPaid

Le service OrderFoodService **reçoit** les messages suivants :

- PayFood

5. DeliveryService

Ce service permet à l'utilisateur de se faire livrer un plat. Un simple message dans indiquant que l'utilisateur effectue une demande de livraison est suffisante.

Le service DeliveryService peut **envoyer** les messages suivants :

- ChooseDeliveryOrder

Le service DeliveryService **reçoit** les messages suivants :

- AskDelivery

Bien entendu, une fois que l'utilisateur décide de faire une demande de livraison, il faudra automatiquement commander et ensuite payer la commande.

Note importante : Il faudra que chaque micro-service soit un *proctype* et ils devront se communiquer entre eux à l'aide de *chan* qui sont des canaux de communication en Promela.

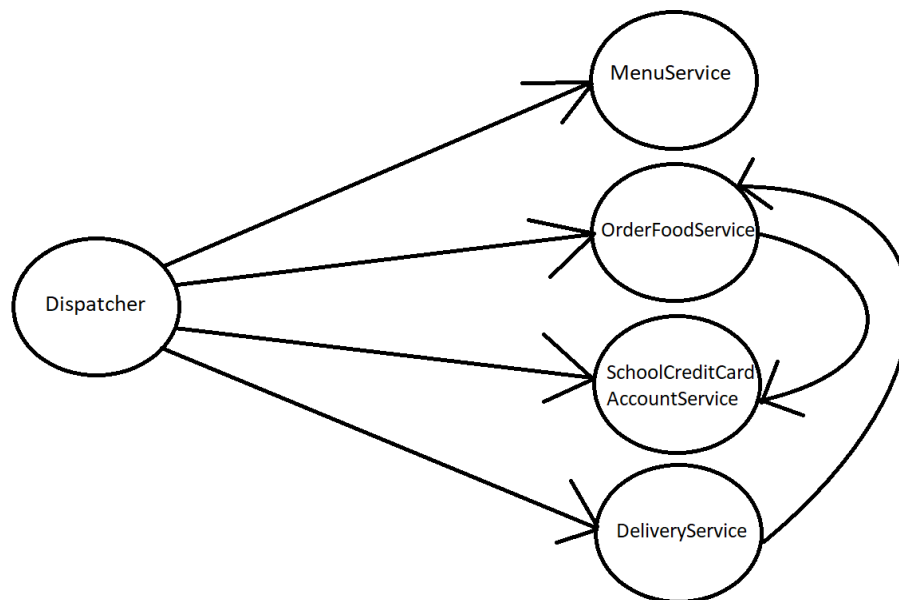


Figure 1 : Représentation des différents services

5. Critères d'évaluation

Gestion adéquate des sections critiques	2 point
Utilisation des canaux pour communiquer entre les processus	1 point
L'ordre d'exécution des processus est adéquat	2 points
Utilisation des goto	-2 points
Code non exécutable	-3 points
Nom de variables et fonctions non spécifiques	Jusqu'à -1 point
Mauvais format de remise	-1 point
Retard	-10 % par jour de retard