

Exercices de modélisation pour le cours de Vérification de systèmes parallèles et logique temporelle

Année 2005-06

1 Enoncé 1

Un processus producteur et n processus consommateurs partagent un buffer limité de b places. Le producteur lit un fichier de nombres entiers qu'il dépose dans le buffer. Les consommateurs retirent les éléments du buffer conformément aux règles suivantes.

- Un seul processus peut accéder au buffer à un instant donné.
- Chaque élément déposé par le producteur doit être reçu par les n consommateurs.
- Chaque consommateur doit recevoir les éléments suivant l'ordre dans lequel ils ont été placés dans le buffer.
- Un consommateur ne peut passer à l'élément suivant que si le producteur l'a introduit et si tous les autres consommateurs ont accédé à l'élément courant.

Dès qu'un processus consommateur reçoit un nombre, il l'affiche accompagné de sa propre identité (un nombre entre 1 et n). Lorsqu'il reçoit le nombre 0, il s'arrête.

Trois paramètres sont passés au programme : le nom du fichier d'entiers, le nombre de processus consommateurs (n) et le nombre de places dans le buffer (b). Dans le fichier, les nombres sont les nombres sont encodés en base 10, sous format ASCII et sont compris entre 0 et 65535 inclus.

Il est demandé d'utiliser les sémaphores et la mémoire partagée (buffer) pour résoudre ce problème.

2 Enoncé 2

Lorsque la transmission d'information par envoi de message est non fiable, ce qui arrive lorsque les messages transitent par un réseau étendu, on utilise des techniques de fiabilisation de la transmission. Ces techniques sont réalisées sous la forme de programmes, appelés protocoles, qui contrôlent la bonne arrivée des messages et retransmettent les messages si nécessaire. Le contrôle de la bonne arrivée d'un message se fait le plus souvent par l'envoi d'un accusé de réception (ou "acquis") du récepteur au transmetteur. Le problème est non seulement d'assurer que tous les messages soient correctement transmis, mais aussi qu'ils ne le soient qu'une seule fois et dans le bon ordre.

Un des protocoles de fiabilisation de la transmission les plus simples est le protocole dit du "bit alterné". Ce protocole ajoute à chaque message un numéro de séquence codé sur un seul bit et donc qui alterne entre 0 et 1. Chaque

message fait l'objet d'un accusé de réception, reprenant le bit de séquence du message reçu. On ne transmet le message suivant que si l'acquis correspondant au message en cours de transmission a été reçu. Si l'acquis n'arrive pas après un délai prédéterminé, le message est retransmis. Comme un message peut être retransmis indûment si son acquis est reçu après un délai trop long, un message reçu est détruit si il a le même numéro que le message précédent. Finalement comme les acquis peuvent aussi se perdre, il y a retransmission de l'acquis si rien n'est reçu après un temps prédéterminé ou si un message avec le même numéro de séquence que le précédent est reçu.

On demande de réaliser une implémentation du protocole du bit alterné entre processus communiquant par files de messages. Pour simuler la perte possible de messages, un processus est placé sur le trajet des messages entre le transmetteur et le récepteur. De manière symétrique, un autre processus est placé sur le trajet des accusés de réception. Ces processus perdront des messages à un taux qui sera fourni en paramètre au programme. Les paquets (message ou accusé de réception) seront perdus de manière aléatoire en respectant le taux donné. Vous utiliserez une méthode de votre choix pour obtenir des valeurs aléatoires.

Pour déclencher les retransmissions après le délai fixé, il faut que le transmetteur et le récepteur puissent recevoir une indication que le délai d'attente est dépassé, ce que l'on appelle usuellement un "time out". Ces conditions de "time out" sont générées par des processus dédiés à cette fonction, un pour le transmetteur et un pour le récepteur. Lors du début du délai d'attente le transmetteur (ou le récepteur) envoie un message à son gestionnaire d'attente qui, le délai étant expiré place un message particulier dans la file d'entrée du transmetteur (ou récepteur).

Les données émises par le processus transmetteur sont simplement la suite des entiers positifs. Le processus récepteur affichera ce qu'il reçoit.

On vous demande d'écrire un programme reproduisant la situation décrite, en exploitant le mécanisme de la file de messages, et d'illustrer le fait que ce protocole est résistant aux pertes de données.

3 Consignes

On demande de modéliser ces systèmes en promela et de vérifier leur validité en utilisant l'outil SPIN disponible à l'adresse suivante :

<http://spinroot.com/spin/whatispin.html>

Bon travail.