

Programmation concurrente

Jackpot

David Gonzalez - Claudio Sousa

1 février 2017

1 Introduction

Ce TP de deuxième année consiste à implémenter une machine à sous de Casino multi-tâches.

1.1 Spécification fonctionnelle

Chaque partie débute avec l'insertion d'une pièce. Ensuite, 3 roues tournent à des vitesses différentes. Chaque roue est arrêtée consécutivement (de gauche à droite) soit manuellement par l'utilisateur, soit automatiquement après 3 secondes.

Lorsque toutes les roues sont arrêtées, le résultat est ajouté à la caisse de la machine. Ce résultat est calculé selon le nombre de chiffres identiques, les résultats possibles sont :

- aucun chiffre identique (perdu) ;
- 2 chiffres identiques (... pièces gagnées) ;
- 3 chiffres identiques (... pièces gagnées).

Ce résultat est affiché pendant 5 secondes, puis revient à l'insertion de la pièce.

1.1.1 Entrées

Les entrées sont gérées à l'aide de signaux générés par la console avec les touches suivantes :

- CTRL-Z (SIGTSTP) : insertion d'une pièce ;
- CTRL-C (SIGINT) : arrêt d'une roue manuellement ;
- CTRL-\ (SIGQUIT) : arrêt du jeu.

1.1.2 Threads

Les threads sont divisés ainsi :

- 1 thread pour le contrôleur du jeu (qui sera le seul à recevoir les signaux) ;
- 1 thread pour l'affichage ;
- N threads, 1 pour chaque roue.

2 Development

2.1 Architecture

FIGURE 1 – Architecture du Jackpot

2.2 Concurrency

FIGURE 2 – Synchronisation

2.3 Méthodologie de travail

2.3.1 Répartition du travail

Ce travail a été effectué à deux.

Nous avons commencés par réfléchir sur papier sur deux éléments :

- architecture du programme : modules et interfaces de bases ;
- premier jet de synchronisation entre les différents threads.

Ensuite, le travail a été réparti ainsi :

- ...

Finalement, nous avons mis en commun les modules et finalisés la synchronisation des threads des différents modules.