Sémaphores

Burkhalter - Lienhard

Version 1.0
4/27/2010 9:04:00 AM

Table des matières

Sémaphore

Cette documentation décrit le programme Sémaphore qui consiste à mettre sur pied un logiciel permettant la représentation d'un barbier, de sa salle d'attente ainsi que des clients ayant recours aux services de celui-ci. Cela a été mis en place dans le laboratoire n°4 du cours PCO.

Introduction

Le but de ce programme est la réalisation d'un programme permettant la simulation d'un salon de coiffure. Une salle d'attente pouvant contenir un certain nombre de clients doit être mise en place. Lorsqu'un client se présente et que la salle d'attente est vide, ou contient encore des sièges non occupés, le client prend place dans la pièce. Dans le cas ou la salle d'attente est pleine, le client va faire un tour et revient tenter sa chance plus tard. Le barbier quant à lui doit couper les cheveux des clients, les uns après les autres. Dans le cas ou il n'est pas entrain de couper les cheveux à un client et que la salle d'attente est vide, il s'endort et sera réveillé par le prochain client à entrer dans la salle d'attente.

Réalisation

La réalisation de notre programme, a nécessité l'implémentation de plusieurs fonctions qui vont être décrites ci-dessous:

Une première fonction 'saisieClavier()' nous permet d'effectuer les saisies utilisateur, c'est à dire le nombre de client ainsi que le nombre de place dans la salle d'attente du barbier.

Une seconde fonction, nommée 'Client()' a été implémentée afin de représenter les clients. Cette fonction sera exécutée par des threads représentant chacun un client. Le nombre de clients, de part la même occasion, de threads utilisant cette fonction est nombre variable dépendant de la saisie utilisateur. Chaque client aura le comportement suivant: ses cheveux poussent; une fois cela fait il se rend chez le barbier et entre dans la salle d'attente. Si la salle d'attente est vide, et donc que le barbier est entrain de dormir, il réveil celui-ci et se fait couper les cheveux. S'il n'est pas le premier dans la salle d'attente et qu'il reste des places de libre, il prend un siège et attend son tour. Finalement dans le cas ou en arrivant il trouve la salle d'attente pleine, il ressort et retentera sa chance après un laps de temps équivalent à la moitié du temps qu'il lui a fallu pour voir ses cheveux pousser.

La dernière fonction que nous avons du implémenter 'Barbier()', permet de représenter le rôle que joue le barbier, et sera donc exécutée uniquement par un thread. De part la fonction, le barbier effectue les actions suivantes: Il regarde si la salle d'attente comporte des clients qui sont prêt à se faire couper les cheveux. Si c'est le cas, il prend le premier afin de lui couper les cheveux et décrémente le nombre de personnes présentes dans la salle. Au contraire, si la salle d'attente est vide, il s'endort jusqu'à ce qu'un client entre dans la salle et le réveil. Cela a été réalisé à l'aide d'une variable booleen 'barbier_endormi', représentant le fait que le barbier soit endormi ou non, ainsi que d'un sémaphore 'barbierDort'.

Tests

Afin de tester que le cahier des charges est respecté et que notre simulateur de salon de coiffure s'opère de manière correcte, nous avons tout d'abord effectué des simulations avec de petits nombres de clients puis avons au fur et à mesure des tests augmenté le nombre de clients présents dans la simulation. Pour chacune de ces phases de tests, nous avons contrôlé la cohérence des résultats obtenus. Pour chacun des tests effectués, les résultats obtenus correspondaient aux valeurs souhaitées. Il est toutefois à noter que lorsqu'un client a fini de se faire couper les cheveux, le client suivant ne passera entre les mains du barbier qu'une fois que le client en cours ait laissé la main à un autre thread.

Index des fichiers

Liste des fichiers

Liste de tous les fichiers documentés avec une brève description :

Documentation des fichiers

Référence du fichier main.c

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <time.h>
#include <semaphore.h>
#include <stdbool.h>
#include <psleep.h>
```

Fonctions

- void * **Client** (void *arg)
- void * **Barbier** (void *arg)
- void **saisieClavier** ()
- int **main** (void)

Variables

- pthread_t **barbier**
- int **nb_clients**
- int **NB_SIEGES**
- int **nb_clients_attente** = 0
- int **dureeCoupe** = 10
- bool **barbier_endormi** = false
- sem_t **barbierDort**
- sem_t **salleAttente**
- sem_t **mutex**
- sem_t **clientDort**
- pthread_t * **tabClients**

Description détaillée

Auteur:
Steve Lienhard et Arnaud Burkhalter

Date:
26.04.2010

Version:
1.0

Description

Ce fichier met en place des threads permettant la gestion de manière concurrente d'un barbier ainsi que des différents clients nécessaires à la à la simulation d'un salon de coiffure.

Documentation des fonctions

void* Barbier (void * *arg*)

But : Fonction qui sera exécutée par un thread traitant le barbier. Une boucle infinie englobe les actions du barbier afin que celles-ci soient exécutées tant que le programme tourne. A chaque itération de la boucle, le barbier va observer si au moins un client est présent dans la salle d'attente. Si c'est le cas, il lui coupe les cheveux, si ce n'est pas le cas, le barbier s'endors et attend que le prochain client le réveil à l'aide du sémaphore 'barbierDors'.

Paramètre(s): arg : pointeur passé à la fonction. Il est inutilisé, mais permet à la fonction de correspondre au prototype nécessaire pour que celle-ci puisse être utilisée par un thread.

void* Client (void * *arg*)

But : Fonction qui sera exécutée par un thread traitant un client. Une boucle infinie englobe les actions du client afin que celles-ci soient exécutées tant que le programme tourne. A chaque itération de la boucle, le client attend une durée de temps aléatoire que ses cheveux poussent puis se rend chez le barbier. Une deuxième boucle while permet au client de tenter ça chance chez le barbier tant qu'il n'a pas réussi à obtenir une place dans la salle attente.

Paramètre(s): arg : pointeur passé à la fonction. Il est utilisé dans notre fonction pour passer le numéro du client. Celui-ci nous est utile pour l'affichage.

int main (void)

But : Fonction principale permettant l'initialisation des sémaphores et la création des threads. Une valeur entière sera retournée, représentant si la fonction s'est terminée de manière correct ou non.

void saisieClavier ()

But : Fonction permettant la saisie de valeur au clavier par l'utilisateur. Le nombre de siège dans la salle d'attente ainsi que le nombre de client sont demandé à l'utilisateur. Les traitements sur les variables sont directement effectués à l'intérieur de la fonction.

Paramètre(s): aucun

Documentation des variables

pthread_t barbier

Thread représentant le barbier

bool barbier_endormi = false

Variable de type bool permettant d'indiquer si le barbier est endormi.

sem_t barbierDort

Sémaphore permettant de représenter le barbier endormi. Le barbier se met en attente sur ce sémaphore lorsque la salle d'attente est vide, et le premier client qui entrera dans la salle d'attente l'incrémentera se qui permet de réveiller le barbier.

sem_t clientDort

Sémaphore représentant un client inactif, état dans lequel il est pendant qu'il se fait couper les cheveux par le barbier. Lorsque le client se fait couper les cheveux, le client décrémente ce sémaphore et le barbier incrémentera celui-ci lorsqu'il aura fini de couper les cheveux du client, permettant ainsi à celui-ci de quitter les lieux.

int dureeCoupe = 10

Variable de type int indiquant le temps nécessaire au barbier pour effectuer une coupe de cheveux.

sem_t mutex

Sémaphore faisant office de mutex et garantissant l'exclusion mutuelle entre les ressources critiques.

int nb_clients

Variable de type int représentant le nombre de clients dans le programme

int nb_clients_attente = 0

Variable de type int représentant le nombre de clients dans la salle d'attente.

int NB_SIEGES

Variable de type int représentant le nombre de place dans la salle d'attente.

sem_t salleAttente

Sémaphore permettant de représenter une liste de clients présents dans la salle d'attente. Chaque client qui prend place dans la salle d'attente décrémente ce sémaphore et lorsque le barbier a fini de traiter son client en cours, il décrémente ce sémaphore et prend ainsi un client afin de lui couper les cheveux. Libérant ainsi une place dans la salle d'attente.

pthread_t* tabClients

Tableau dynamique permettant de contenir un nombre variable de clients. Un nombre de threads variable sera contenu dans tableau. Ce nombre sera égal au nombre saisi par l'utilisateur en début d'exécution du programme.

Index