Moniteurs

Burkhalter - Lienhard

Version 1.0
6/15/2010 12:00:00 PM

Table des matières

Moniteurs

Cette documentation décrit le programme Moniteurs qui doit permettre la simulation d'un salon de coiffure à l'aide de moniteurs. Cela a été mis en place dans le laboratoire n°7 du cours PCO.

Introduction

Le but de ce programme est la réalisation d'un programme permettant la simulation d'un salon de coiffure et tatouage. 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 s'occuper de l'un des clients 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.

Realisation

Afin de réaliser ce programme à l'aide de moniteur, nous avons le plus possible exploiter le programme réalisé un mois auparavant afin de ne pas réinventer la roue. Nous avons tout d'abord créer le moniteur qui va permettre de gérer les accès concurents aux différentes variables. Ce moniteur est composé d'une fonction initialiserTampon qui permet d'initialiser correctement les différents mutex, d'une fonction détruiteTampon qui permet de libérer la mémoire, d'une fonction entreeSalle permettant de gérer l'accès des clients à la salle d'attente, d'une fonction couperCheveux qui permet de gérer la coupe de cheveux des clients. Les variables conditions utilisées sont les variables : clientDort, barbierDort. Nous avons ensuite deux fonctions, Barbier et Client qui permettent la gestion du barbier et des différents clients. La fonction barbier est exécutée par le thread du barbier et la fonction client par les différents clients du programme Les explications relatives à ces deux fonctions sont décrites dans les en-têtes respectifs des deux fonctions.

monitor.c Ce fichier contient l'implémentation des fonctions définies dans **monitor.h** Les différentes fonctionnalités réalisées par les fonctions sont décrites dans les en-têtes respectives des fonctions, situées dans le fichier **monitor.h**

monitor.h Ce fichier contient les en-têtes et les prototypes des différentes fonctions du moniteur.

monitor.c Ce fichier contient l'implémentation des fonctions définies dans **monitor.h** Nous avons donc la fonction entreeSalle qui prend en paramètre la durée de pousse des cheveux ainsi que le numéro du client exécutant la fonction. Cette fonction gère l'accès à la salle d'attente de la manière suivante : Tant que la salle d'attente a un ou plusieurs sièges de libres, le client se met en attente sur &salleAttente (variable condition du moniteur). Si par contre la salle est pleine, le client ressort et attend une durée déterminée avant de ressayer.

Conlusion

Bien qu'ayant eu quelques difficulté avec la notion de moniteur vue en théorie, nous sommes parvenus à nous documents suffisamment sur le sujet afin de pouvoir répondre au cahier des charges de ce laboratoire. Après les différents tests effectués, nous pouvons affirmer que notre laboratoire est fonctionnel. La conception à l'aide de moniteur assure donc une meilleure stabilité et une meilleure structure du programme que la simple utilisation de sémaphores, même si la plupart des choses restent faisables sans ce concept.

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>
#include "monitor.h"
```

Fonctions

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

Variables

- pthread_t **barbier**
- pthread_t * **tabClients**
- int **DUREE_COUPE** = 3
- int **NB_SIEGES**
- int **NB_CLIENTS**

Description détaillée

Auteur:

Burkhalter Arnaud et Lienhard Steve

Date:

15.06.2010

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 simulation d'un salon de coiffure, et cela à l'aide de MONITEURS.

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 de la condition '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.

Paramï¿½tre(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 l'intérieur de la fonction.

Paramètre(s): aucun

Documentation des variables

pthread_t barbier

Thread représentant le barbier

int DUREE_COUPE = 3

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

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