

Laboratoire de programmation concurrente (PCO1) semestre printemps 2009 - 2010

Prise en main des threads POSIX

Temps à disposition : 6 périodes (travail débutant en semaine 1)

1 Objectifs pédagogiques

- Se familiariser avec un environnement de programmation ;
- Mettre en évidence le problème de l'exclusion mutuelle ;
- Réaliser son premier programme concurrent en C.

2 Cahier des charges

Nous désirons réaliser un logiciel pour la gestion d'un téléphone public. Ce téléphone dispose d'un combiné qui peut être décroché ou non, de 10 touches permettant d'entrer un numéro, d'un lecteur de carte à puce, et d'une fente permettant d'insérer de la monnaie. Le but de ce laboratoire est de gérer le fonctionnement du téléphone selon les contraintes suivantes :

1. Si une carte est présente, la monnaie entrée est directement créditée sur la carte, sinon elle est immédiatement rendue ;
2. Une carte de téléphone doit être insérée avant que le combiné ne soit décroché pour qu'une communication puisse avoir lieu ;
3. La composition du numéro n'est prise en compte que lorsque le combiné est décroché ;
4. Un numéro de téléphone est composé de 10 chiffres, ni plus ni moins ;
5. Lorsque le numéro de téléphone est composé et que les conditions précédentes sont remplies, la communication a lieu ;
6. Durant la communication, le coût débité de la carte est de une unité toutes les deux secondes ;
7. La communication se termine si l'utilisateur raccroche le téléphone ou retire la carte, ou si le crédit de la carte arrive à 0 ;
8. Attention à gérer correctement le fait que l'utilisateur pourrait enlever et remettre une carte avant que la communication ne démarre.

3 Travail à faire

Pour réaliser le travail demandé il sera notamment nécessaire de gérer la quantité de crédit restant sur la carte. En utilisant le simulateur *telephone* qui vous est fourni, réalisez le programme énoncé en vous inspirant de l'exemple fourni en annexe. Les différentes actions de l'utilisateur se font par le clavier. Les chiffres allant de 0 à 9 simulent l'appui des touches de numérotation du téléphone. Les touches de la ligne supérieure du clavier, à savoir qwertzuiop simulent l'introduction de pièces de valeur 1 à 9. La touche 'I' (majuscule) simule l'introduction de la carte, et la touche 'O' (majuscule) le fait qu'elle est retirée par l'utilisateur. La touche 'D' simule le fait de décrocher le combiné, et la touche 'R' celui de le raccrocher. L'interface de ce simulateur vous est aussi fournie en annexe. Elle comprend :

1. Deux fonctions bloquantes qui signalent respectivement l'introduction d'une carte et sa récupération.
2. Deux fonctions bloquantes qui attendent respectivement que le combiné soit décroché et raccroché.
3. Une fonction bloquante qui attend qu'une touche de numérotation soit pressée.
4. Une fonction bloquante qui attend qu'une pièce soit introduite dans la machine.
5. Une fonction pour modifier le montant disponible sur la carte.
6. Deux procédures permettant de gérer une section critique.
7. Une fonction d'initialisation du téléphone, qui doit être appelée en début de programme.

Le code de l'interface, un exemple, ainsi que la documentation de l'interface sont disponibles sur

<http://reds.heig-vd.ch/Formations/Bachelor/PC01/PC01doc.aspx>

Il n'est pas permis de modifier le simulateur.

4 Travail à rendre

- Vous devez nous rendre un listage complet de vos sources et aussi nous les transmettre par courrier électronique. Vous devez en outre fournir, via courrier électronique, un fichier pdf créé sur la base de ce que doxygen a généré pour votre documentation.
- La description de l'implémentation, ses différentes étapes, la manière dont vous avez vérifié son fonctionnement et toute autre information pertinente doivent figurer dans les programmes rendus, et devrait ressortir via la documentation générée par doxygen. Aucun rapport n'est demandé.
- Inspirez-vous du barème de correction pour connaître là où il faut mettre votre effort.
- Vous pouvez travailler en équipe de deux personnes au plus.

5 Barème de correction

Conception	15%
Exécution et fonctionnement	20%
Tests	10%
Codage	15%
Documentation et en-têtes des fonctions	30%
Commentaires au niveau du code	5%
Robustesse	5%