

Sémaphore local

Anthony Araye et Camille Schnell

21 février 2018

Sommaire

1	Introduction	2
2	Manuel d'utilisation	3
2.1	Tutoriel d'utilisation	3
2.2	initialize	4
2.3	acquire	5
2.4	release	6
2.5	destroy	7
3	Implémentation	8
4	Recette	8

1 Introduction

Cinq philosophes se trouvent autour d'une table avec en face d'eux un plat de spaghetti et à gauche de chaque plat se trouve un couvert (une fourchette ou un couteau). Un philosophe ne possède que trois états :

- penser pendant un temps indéterminé*
- être affamé pendant un temps déterminé et fini*
- manger pendant un temps déterminé et fini.*

Cependant, quand un philosophe a faim, il se met en état "affamé" et va attendre que les couverts autour de son assiette soient libres pour pouvoir manger. Et dans le cas où l'un des deux couverts n'est pas libre, le philosophe se met en état de famine pendant un temps déterminé en attendant de révéifier.

Cette situation représente en réalité le problème du "dîner des philosophes" énoncé par Dijkstra.

Le but de ce sujet est d'implémenter un système de sémaphore afin de gérer des sections critiques et ainsi répondre au problème ci-dessus.

2 Manuel d'utilisation

2.1 Tutoriel d'utilisation

2.2 initialize

Nom

`sem_initialize` - Initialise un sémaphore

Synopsis

```
#include <sem.h>
int sem_initialize(sem *sem, int value)
```

Description

`sem_initialize()` alloue et initialise *sem* avec *value* comme valeur initiale. *value* doit être supérieur ou égale à 0.

Valeur renvoyée

`sem_initialize()` renvoie 0 dans le cas où le sémaphore a bien été créé, -1 si une erreur a été levée.

Erreurs

EFAULT : *value* est strictement inférieur à 0.

2.3 acquire

Nom

Synopsis

Description

Valeur renvoyée

Erreurs

2.4 release

Nom

Synopsis

Description

Valeur renvoyée

Erreurs

2.5 destroy

Nom

Synopsis

Description

Valeur renvoyée

Erreurs

3 Implémentation

4 Recette