

# Utiliser les sémaphores pour synchroniser des threads

---

**Attention** : il est conseillé de lire la documentation sur les *threads* avant cette page<sup>1</sup>.

## Description des sémaphores

La librairie de gestion des *threads* offre les fonctions ci-dessous pour créer et utiliser des sémaphores. **Attention** : ces sémaphores sont propres à un processus. Ils permettent de synchroniser plusieurs threads entre eux, mais ils ne peuvent synchroniser plusieurs processus. Pour réaliser cette synchronisation il faut se tourner vers les sémaphores système V<sup>2</sup> basés sur les IPC (*Inter Processus Communication*).

`int sem_init(sem_t *semaphore, int pshared, unsigned int valeur)` Création d'un sémaphore et préparation d'une valeur initiale.

`int sem_wait(sem_t * semaphore)` ; Opération P sur un sémaphore.

`int sem_trywait(sem_t * semaphore)` ; Version non bloquante de l'opération P sur un sémaphore.

`int sem_post(sem_t * semaphore)` ; Opération V sur un sémaphore.

`int sem_getvalue(sem_t * semaphore, int * sval)` ; Récupérer le compteur d'un sémaphore.

`int sem_destroy(sem_t * semaphore)` ; Destruction d'un sémaphore.

Bien entendu, une aide plus complète peut être obtenue sur chaque fonction en utilisant le manuel UNIX :

`man nom_de_la_fonction`

## Un exemple d'utilisation des sémaphores

Cet exemple illustre la mise en oeuvre d'une section critique (mutuelle exclusion) permettant d'éviter un mélange des affichages réalisés par les deux threads.

---

<sup>1</sup>ref:dil-doc-thread

<sup>2</sup>ref:dil-doc-IPC

```

#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <semaphore.h>
#include <unistd.h>

sem_t mutex;

void* affichage (void* name)
{
    int i, j;
    for(i = 0; i < 20; i++) {
        sem_wait(&mutex); /* prologue */
        for(j=0; j<5; j++) printf("%s ",(char*)name);
        sched_yield(); /* pour etre sur d'avoir des problemes */
        for(j=0; j<5; j++) printf("%s ",(char*)name);
        printf("\n ");
        sem_post(&mutex); /* epilogue */
    }
    return NULL;
}

int main (void)
{
    pthread_t filsA, filsB;

    sem_init(&mutex, 0, 1);

    if (pthread_create(&filsA, NULL, affichage, "AA")) {
        perror("pthread_create");
        exit(EXIT_FAILURE);
    }
    if (pthread_create(&filsB, NULL, affichage, "BB")) {
        perror("pthread_create");
        exit(EXIT_FAILURE);
    }

    if (pthread_join(filsA, NULL))
        perror("pthread_join");

    if (pthread_join(filsB, NULL))
        perror("pthread_join");

    printf("Fin du pere\n") ;
    return (EXIT_SUCCESS);
}

```

Pour compiler et effectuer l'édition de liens vous devez utiliser la ligne suivante :

```
cc votre_programme.c -lpthread -o votre_programme
```

c'est la fin !