

# Parallélisme à base de thread

(PBT 2022-2023)

## TD3

Modèle de programmation Pthread

[hugo.taboada@cea.fr](mailto:hugo.taboada@cea.fr)

[marc.perache@cea.fr](mailto:marc.perache@cea.fr)



Click here or scan to download TD'files from pcloud link.

## I Sémaphore

### 1 Exemple

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <pthread.h>
4 #include <semaphore.h>
5 #include <unistd.h>
6
7 sem_t mutex;
8
9 void *affichage (void*name){
10     int i, j;
11     for (i=0; i<20; i++){ //nombre de ligne
12         sem_wait(&mutex);
13         for (j=0; j<5; j++){ //Affiche 5 char
14             printf ("%s", (char*)name);
15         }
16         sched_yield();
17         for (j=0; j<5; j++){ //Affiche 5 char
18             printf ("%s", (char*)name);
19         }
20         printf ("\n");
21         sem_post(&mutex);
22     }
23     return NULL;
24 }
25 int main(void){
26     pthread_t filsA, filsB;
27     sem_init(&mutex, 0, 1);
28     if (pthread_create(&filsA, NULL, affichage, "A")){
29         perror("pthread_create");
30         exit(EXIT_FAILURE);
31     }
32     if (pthread_create(&filsB, NULL, affichage, "B")){
33         perror("pthread_create");
34         exit(EXIT_FAILURE);
35     }
```

```

36     if(pthread_join(filsA ,NULL)){
37         perror("pthread_join");
38         exit(EXIT_FAILURE);
39     }
40     if(pthread_join(filsB ,NULL)){
41         perror("pthread_join");
42         exit(EXIT_FAILURE);
43     }
44     return EXIT_SUCCESS;
45 }

```

**Q.1:** Décrire à l'aide d'un schéma l'utilité du sémaphore dans l'exemple précédent.

## II Mise en place des sémaphores dans mthread

**Q.2:** Mettre en place la fonction `mthread_sem_init`

**Q.3:** Mettre en place la fonction `mthread_sem_wait`

**Q.4:** Mettre en place la fonction `mthread_sem_post`

**Q.5:** Mettre en place la fonction `mthread_sem_destroy`

**Q.6:** Mettre en place la fonction `mthread_sem_trywait`

**Q.7:** Mettre en place la fonction `mthread_sem_getvalue`

## III Démonstration

**Q.8:** Pour chacune des fonctions précédentes, construire un programme d'exemple qui teste leur bon fonctionnement.

Ce TD est noté. Le code source ainsi qu'un rapport répondant aux questions et **détaillant l'implémentation** est à envoyer avant le 12/03/2023 à 23h59 sur le lien pcloud suivant :

<https://e.pcloud.com/#page=puplink&code=P3wZAIwGD9eYS6L3ymVsJ6I7pJ49nlPy>



Pour toutes questions n'hésitez pas à m'envoyer un mail à l'adresse suivante :  
*hugo.taboada@cea.fr*