



UNIVERSITÉ SIDI MOHAMED BEN ABDLLAH
FACULTÉ DES SCIENCES DHAR EL MAHRAZ DE FÈS



DÉPARTEMENT D'INFORMATIQUE

Compte Rendu : TP 8



Réaliser par :

Nom	CNE	Gr
Dahman Iliass	S147045000	1
El Khalfaoui Mohammed	S136327808	2

Enseignant :

Mohammed Meknassi

TP : Mohammed Achbar

EXERCICE 1:

Ecrire un programme qui crée deux threads dont chacune appelle la même fonction nommée 'doSomething'. Cette dernière imprime au début de son exécution un message du commencement de l'exécution ainsi que la valeur de la variable 'counter' qui correspond à un compteur. Après cette impression, la fonction effectue un certain traitement et imprime la fin de son exécution par un message accompagné de la valeur du 'counter'. Les deux fonctions partagent la variable 'counter'. Cette variable s'incrémente chaque fois que 'doSomething' est appelé par l'un ou l'autre des deux threads. Que remarquez-vous ?

Corrigé :

```
mohammed@khalifaoui-HP-EliteBook-820-G1:~/TPSYS/TP8$ ./ex2
Au debut   : 1
Au fin     : 2
Au debut   : 2
Au fin     : 2
mohammed@khalifaoui-HP-EliteBook-820-G1:~/TPSYS/TP8$
```

```

#include<stdio.h>
#include<pthread.h>
#include<string.h>
#include<stdlib.h>
int compteur=0;
void *thread_1()
{
    int i;
    int s=0;
    compteur++;
    printf("Au debut  : %d\n",compteur);

    for(i=0;i<1999;i++)
        s++;

    printf("Au fin    : %d\n",compteur);
}
int main ()
{
    compteur=0;
    pthread_t  th1,th2;
    pthread_create(&th1,NULL,thread_1,NULL);
    pthread_create(&th2,NULL,thread_1,NULL);

    pthread_join(th1,NULL);
    pthread_join(th2,NULL);
    return 0;
}

```

EXERCICE 2:

Modifier le programme de l'exercice 1 en introduisant les fonctions de synchronisation les mutex afin d'afficher les bons résultats.

Corrigé :

```
#include<stdio.h>
#include<pthread.h>
#include<string.h>
#include<stdlib.h>
int compteur;
pthread_mutex_t lock=PTHREAD_MUTEX_INITIALIZER;
void *thread_1()
{ pthread_mutex_lock(&lock);
  int i;
  int s=0;
  compteur++;
  printf("Au debut : %d\n",compteur);

  for(i=0;i<1999;i++)
    s++;

  printf("Au fin : %d\n",compteur);
  pthread_mutex_unlock(&lock);
}
int main ()
{

  compteur=0;
  pthread_t th1,th2;
  pthread_create(&th1,NULL,thread_1,NULL);
  pthread_create(&th2,NULL,thread_1,NULL);

  pthread_join(th1,NULL);
  pthread_join(th2,NULL);
  return 0;
}
```

File Edit View Search Terminal Help

mohammed@khalfaoui-HP-EliteBook-820-G1:~/TPSYS/TP8\$./Ex1

Au debut : 1

Au fin : 1

Au debut : 2

Au fin : 2

mohammed@khalfaoui-HP-EliteBook-820-G1:~/TPSYS/TP8\$

EXERCICE 3:

En utilisant les fonctions de synchronisation de mutex, écrire un programme C qui crée deux threads qui appellent deux fonctions (a) 'write_tab_process' et (b) 'read_tab_process'. La 1ère fonction (a) remplit un tableau tab de 10 éléments dont chaque case contient le double de son indice ($2*i$). La 2ème fonction (b) affiche le contenu de tab sur la sortie standard. Il est à préciser que les deux threads partagent entre elles le tableau tab.

Corrigé :

```
#include<stdio.h>
#include<pthread.h>
#include<unistd.h>
int T[10];
int i;
pthread_mutex_t lock;
void * write_tab()
{
pthread_mutex_lock(&lock);
    for(i=0;i<10;i++)
    {
        T[i]=2*i;
    }
pthread_mutex_unlock(&lock);
}
void * read_tab()
{
pthread_mutex_lock(&lock);
    for(i=0;i<10;i++)
        printf("Element numero %d : %4d\n",i+1,T[i]);

pthread_mutex_unlock(&lock);
}
int main()
{
pthread_mutex_init(&lock,NULL);
pthread_t th1,th2;
pthread_create(&th1,NULL,write_tab,NULL);
pthread_create(&th2,NULL,read_tab,NULL);
pthread_join(th1,NULL);
pthread_join(th2,NULL);
pthread_mutex_destroy(&lock);
return 0;
}
```

File Edit View Search Terminal Help

mohammed@khalfaoui-HP-EliteBook-820-G1:~/TDSYS/thread\$./ex3

Element numero 1 : 0

Element numero 2 : 2

Element numero 3 : 4

Element numero 4 : 6

Element numero 5 : 8

Element numero 6 : 10

Element numero 7 : 12

Element numero 8 : 14

Element numero 9 : 16

Element numero 10 : 18

mohammed@khalfaoui-HP-EliteBook-820-G1:~/TDSYS/thread\$