



جامعة سيدي محمد بن عبد الله
كلية العلوم ظهر المهرز

جامعة سيدي محمد بن عبد الله
كلية العلوم ظهر المهرز

Université Sidi Mohamed Ben Abdellah
Faculté des Sciences Dhar Mahraz



Compte Rendu De Travaux Pratique

System d'Exploitation

Travaille Pratique 7 : Les Threads



SMI – S4

Réaliser par :

Prénom :	CNE :	Nom :
Achraf	1513755449	Tazi
Oussama	1311778906	Nadrani

Exercice1 :

(a)Quelle bibliothèque devrait être incluse dans mon programme c de gestion des threads ?

`<pthread.h>`

(b) Comment compiler un programme nommé «myhello.c» en utilisant les threads POSIX de linux ?

```
gcc -pthread '/home/kamal/myhello.c' -o  
verif
```

Exercice2 :

Ecrire un programme c qui crée deux threads dont le 1er appel la fonction task_a et le 2ème appel la fonction task_b. Task_a boucle 5 fois en affichant à chaque boucle « hello Word A » suivi du numéro de la boucle. Task_b boucle 7 fois et affiche « hello Word B » et le numéro de la boucle. Le main commence par l’affichage du message "main init" (avant la création des deux threads) et finit par l’affichage du message "main end" (après la création des threads). Que remarquez-vous comme résultat affiché? Ajoutez au programme la fonction pthread_join pour chaque création du thread. Que remarquez-vous ?

```

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

static void *task_a(void *p_data){
    int i;
    printf("\n");
    for(i=0;i<5;i++){
        printf("Hello World %d \n", i);
    }
    (void) p_data;
    return NULL;
}

static void *task_b(void *p_data){
    int i;
    for(i=0;i<7;i++){
        printf("job_%d\t", i);
    }
    (void) p_data;
    return NULL;
}

```

```

int main(){
    pthread_t ta;
    pthread_t tb;
    printf("main start ");

    pthread_create(&ta,NULL,task_a,NULL);
    pthread_create(&tb,NULL,task_b,NULL);
    #if 1
    pthread_join(ta,NULL);
    pthread_join(tb,NULL);
    #endif
    printf("main end");
    return 0;
}

```

```

nadrani@nadrani-pc:~/sys_tp/TpSys7$ gcc -pthread travail1.c -o TP7_1
nadrani@nadrani-pc:~/sys_tp/TpSys7$ ./TP7_1
main start job_0      job_1  job_2  job_3  job_4  job_5  job_6
Hello World 0
Hello World 1
Hello World 2
Hello World 3
Hello World 4

```

Exercice3 :

Ecrire un programme C qui crée sept threads (avec un test sur l'erreur de création) dont chacun affiche le message « Hello World! » avec son numéro et, termine avec pthread_exit().

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<string.h>
#include<pthread.h>

void *fonction_threads(void *thid){
    float tid;
    tid = (long)thid;
    printf("Hello World! je suis thread %ld\n", (long)tid);
    pthread_exit(NULL);
}

int main(){
    pthread_t thread[7];
    int rc;
    long i;
    for(i=0; i<7; i++){
        printf("Creation thread %ld\n", i);
        rc = pthread_create(&thread[i],NULL,fonction_threads,(void*)i);
    }
    if(rc){
        printf("Error: code d'erreur de pthread_create() est %d",rc);
        exit(0);
    }
    pthread_exit(NULL);
}
```

Resultat

```
nadrani@nadrani-pc:~/sys_tp/TpSys7$ gcc -pthread travail2.c -o TP7_2
nadrani@nadrani-pc:~/sys_tp/TpSys7$ ./TP7_2
Creation thread 0
Creation thread 1
Creation thread 2
Creation thread 3
Creation thread 4
Creation thread 5
Creation thread 6
Hello World! je suis thread 6
Hello World! je suis thread 5
Hello World! je suis thread 4
Hello World! je suis thread 3
Hello World! je suis thread 2
Hello World! je suis thread 1
Hello World! je suis thread 0
```