



ASSIGNMENT # 4

NAME : SHAHERYAR ASHFAQ

ROLL NO : **20P-0128**

SECTION : BS-CS **4-B**

SUBJECT : OPERATING SYSTEM

QUESTION 1

```
C q1_A4.c > main()
1  #include <stdio.h>
2  #include <pthread.h>
3  #include <stdlib.h>
4  void * thread1(){
5
6      for(int c=0 ; c<10 ; c++){
7          printf("Hello\n");
8      }
9  }
10 void * thread2(){
11
12     for(int c=0 ; c<10 ; c++){
13         printf("World\n");
14     }
15 }
16 int main(){
17     int status;
18     pthread_t tid1,tid2,tid3,tid4;
19
20     pthread_create(&tid1,NULL,thread1,NULL);
21     pthread_create(&tid3,NULL,thread1,NULL);
22
23     pthread_join(tid1,NULL);
24
25
26     pthread_join(tid3,NULL);
27     pthread_create(&tid2,NULL,thread2,NULL);
28
29     pthread_create(&tid4,NULL,thread2,NULL);
30
31     pthread_join(tid4,NULL);
32     pthread_join(tid2,NULL);
33
34     return 0;
35
36 }
```

OUTPUT WITH 2 THREADS

```
shaheryar@ubuntu:~/LAB TASKS$ gcc -lpthread q1_A4.c -o q1_A4  
shaheryar@ubuntu:~/LAB TASKS$ ./q1_A4
```

```
Hello  
Hello  
Hello  
Hello  
Hello  
Hello  
Hello  
Hello  
Hello  
Hello  
World  
World  
World  
World  
World  
World  
World  
World  
World  
World  
World
```



```
shaheryar@ubuntu:~/LAB TASKS$
```

OUTPUT WITH 4 THREADS

[illegible]

OUTPUT EXPLANATION

The main thread waits till the newly created thread exits. Therefore, the final line of the output is printed only after the new thread exits.

The thread scan terminate independently of each other by not using the `pthread_join` function.

C q1_A4.c C a2_a4.c X C test.c

```
C a2_a4.c > main()
1  #include <unistd.h>
2  #include <sys/types.h>
3  #include <errno.h>
4  #include <stdlib.h>
5  #include <pthread.h>
6  #include <string.h>
7  #include <stdio.h>
8
9  #define NUM_RUNS 10000000
10
11 void handler (void *ptr);
12
13 int counter;
14
15 int main(){
16     int i[2];
17     pthread_t thread_a;
18     pthread_t thread_b;
19
20     i[0] = 0;
21     i[1] = 1;
22
23     pthread_create ( &thread_a, NULL, (void *) &handler, (void *) &i[0]);
24     pthread_create ( &thread_b, NULL, (void *) &handler, (void *) &i[1]);
25
26     pthread_join( thread_a, NULL);
27     pthread_join( thread_b, NULL);
28
29     printf("-----\n");
30     printf("Final counter value: %d\n", counter);
31     printf("Error: %d\n", (NUM_RUNS*2-counter));
32     exit(0);
33
34 }
35
36 void handler(void *ptr){
37     int iter=0;
38     int thread_num;
39     thread_num=((int*)ptr);
40     printf("Starting Thread: %d \n",thread_num);
41
42     while(iter<NUM_RUNS){
43         counter++;
44         iter +=1;
45     }
46     printf("Thread %d,counter = %d \n",thread_num,counter);
47     pthread_exit(0);
48 }
```

```
Activities Terminal Apr 1 21:37 shaheryar@ubuntu: ~/LAB TASKS
shaheryar@ubuntu:~/LAB TASKS$ gcc -lpthread a2_a4.c -o a4
shaheryar@ubuntu:~/LAB TASKS$ ./a4
Starting Thread: 0
Starting Thread: 1
Thread 1,counter = 11806262
Thread 0,counter = 14956756
-----
Final counter value: 14956756
Error: 5043244
shaheryar@ubuntu:~/LAB TASKS$ gcc -lpthread a2_a4.c -o a4
shaheryar@ubuntu:~/LAB TASKS$ ./a4
Starting Thread: 0
Starting Thread: 1
Thread 1,counter = 12545593
Thread 0,counter = 13366049
-----
Final counter value: 13366049
Error: 6633951
shaheryar@ubuntu:~/LAB TASKS$ time ./a4 >/dev/null

real    0m0.093s
user    0m0.175s
sys     0m0.001s
shaheryar@ubuntu:~/LAB TASKS$
```

Q.1 What should be the value of the counter variable at the end?

Ans 14956756

Q.2 What is the value you get?

Thread 0 Counter = 14956756

Thread 1 Counter = 11806262

Q.3 How large is the error and how much does it vary on different runs?

Ans: Error Value = 5043244.

Variance = 1,590,707

Q.4 How much user time (roughly) does the program take to run on your system?

Ans : 0.175s