

Subhan Khalid

P200086 BSCS 4A



Q1.1:

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

OUTPUT

```
abdul@abdul-HP-EliteBook-840-G4:~/Documents/BSCS-4_semester/Operating Systems$ gcc Assignment_04_Q1.c -o Assignment_04_Q1 -lpthread
abdul@abdul-HP-EliteBook-840-G4:~/Documents/BSCS-4_semester/Operating Systems$ ./Assignment_04_Q1
World
World
World
World
World
World
World
World
World
World
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
abdul@abdul-HP-EliteBook-840-G4:~/Documents/BSCS-4_semester/Operating Systems$
```

Q1.2:-Updated Program

The screenshot shows a Visual Studio Code editor window with the title bar "Assignment_04_Q1.c - Operating Systems - Visual Studio Code". The editor is open to a file named "Assignment_04_Q1.c". The code is a C program that demonstrates thread creation and joining. It includes headers for `stdio.h`, `pthread.h`, and `stdlib.h`. It defines two thread functions, `thread1` and `thread2`, each of which prints a message and then exits. The `main` function creates four threads (two of each type) and joins them before returning 0.

```

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

```

The status bar at the bottom indicates the current line and column (Ln 31, Col 14), the number of spaces (4), the encoding (UTF-8), the line ending (LF), and the operating system (Linux). The window title bar also shows the date and time (Apr 1, 2:57 PM) and the battery level (78%).

OUTPUT

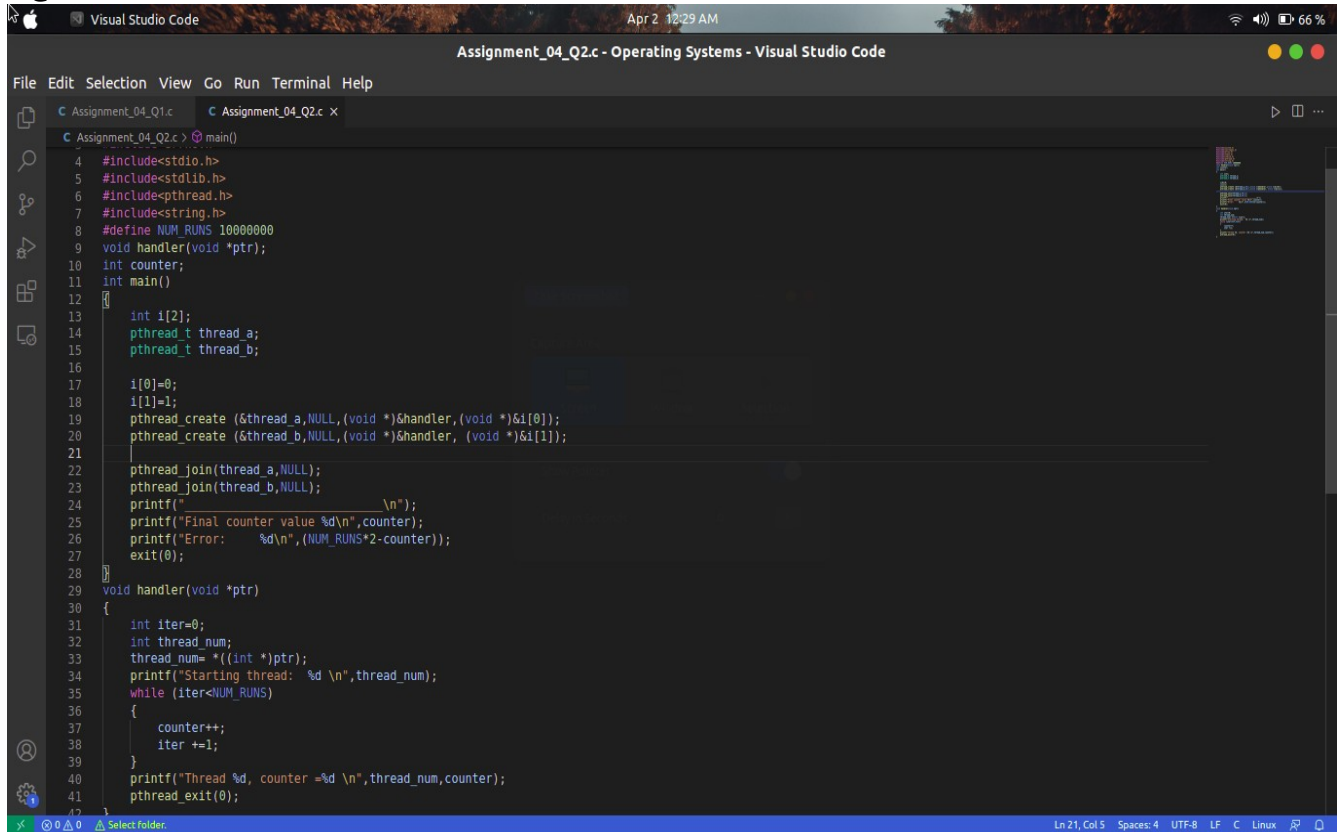
```

abdul@abdul-HP-EliteBook-840-G4:~/Documents/BSCS-4_semester/Operating Systems$ gcc Assignment_04_Q1.c -o Assignment_04_Q1 -lpthread
abdul@abdul-HP-EliteBook-840-G4:~/Documents/BSCS-4_semester/Operating Systems$ ./Assignment_04_Q1
Hello
Hello
Hello
Hello
Hello
Hello
Hello
World
World
World
Hello
World
World
Hello
World
World
World
Hello
Hello
Hello
World
World
Hello
World
World
World
Hello
Hello
Hello
World
World
World
World
World
Hello
Hello
World
World
World
Hello

```

```
abdul@abdul-HP-EliteBook-840-G4:~/Documents/BSCS-4_semester/Operating Systems$ gcc Assignment_04_Q1.c -o Assignment_04_Q1 -lpthread
abdul@abdul-HP-EliteBook-840-G4:~/Documents/BSCS-4_semester/Operating Systems$ ./Assignment_04_Q1
Hello
Hello
Hello
Hello
World
World
World
World
World
World
World
World
World
World
World
World
World
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
World
World
World
World
World
World
World
World
World
```

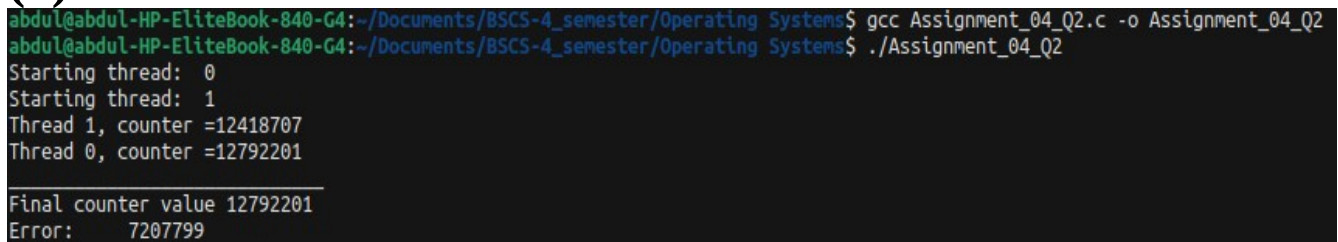
Q2:-



```
1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<pthread.h>
4  #include<string.h>
5  #define NUM_RUNS 1000000
6  void handler(void *ptr);
7  int counter;
8  int main()
9  {
10     int i[2];
11     pthread_t thread_a;
12     pthread_t thread_b;
13
14     i[0]=0;
15     i[1]=1;
16     pthread_create (&thread_a,NULL,(void *)&handler,(void *)&i[0]);
17     pthread_create (&thread_b,NULL,(void *)&handler, (void *)&i[1]);
18
19     pthread_join(thread_a,NULL);
20     pthread_join(thread_b,NULL);
21     printf("\n");
22     printf("Final counter value %d\n",counter);
23     printf("Error:  %d\n", (NUM_RUNS*2-counter));
24     exit(0);
25 }
26 void handler(void *ptr)
27 {
28     int iter=0;
29     int thread_num;
30     thread_num= *((int *)ptr);
31     printf("Starting thread:  %d \n",thread_num);
32     while (iter<NUM_RUNS)
33     {
34         counter++;
35         iter +=1;
36     }
37     printf("Thread %d, counter =%d \n",thread_num,counter);
38     pthread_exit(0);
39 }
```

OUTPUT

(a)



```
abdul@abdul-HP-EliteBook-840-G4:~/Documents/BSCS-4_semester/Operating Systems$ gcc Assignment_04_Q2.c -o Assignment_04_Q2
abdul@abdul-HP-EliteBook-840-G4:~/Documents/BSCS-4_semester/Operating Systems$ ./Assignment_04_Q2
Starting thread:  0
Starting thread:  1
Thread 1, counter =12418707
Thread 0, counter =12792201

Final counter value 12792201
Error:  7207799
```

(b)

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

Ans:

The value of counter variable should be 2 times of “NUM_RUNS” because “NUM_RUNS” variable call 2 times in the code

ii. What is the value you get?

Ans:

Value = 12792201

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

Ans:

- (i) Error is in Millions
- (ii) Error varies in lac's

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

Ans:

usr = 0.124s

```
abdul@abdul-HP-EliteBook-840-G4:~/Documents/BSCS-4_semester/Operating Systems$ time ./Assignment_04_Q2
Starting thread: 0
Starting thread: 1
Thread 1, counter =8719841
Thread 0, counter =11558715
Final counter value 11558715
Error:      8441285
real      0m0.067s
user      0m0.124s
sys       0m0.000s
abdul@abdul-HP-EliteBook-840-G4:~/Documents/BSCS-4_semester/Operating Systems$
```