

ASSIGNMENT # 03

NAME: BASHARAT HUSSAIN

ROLL NO: P17-6102

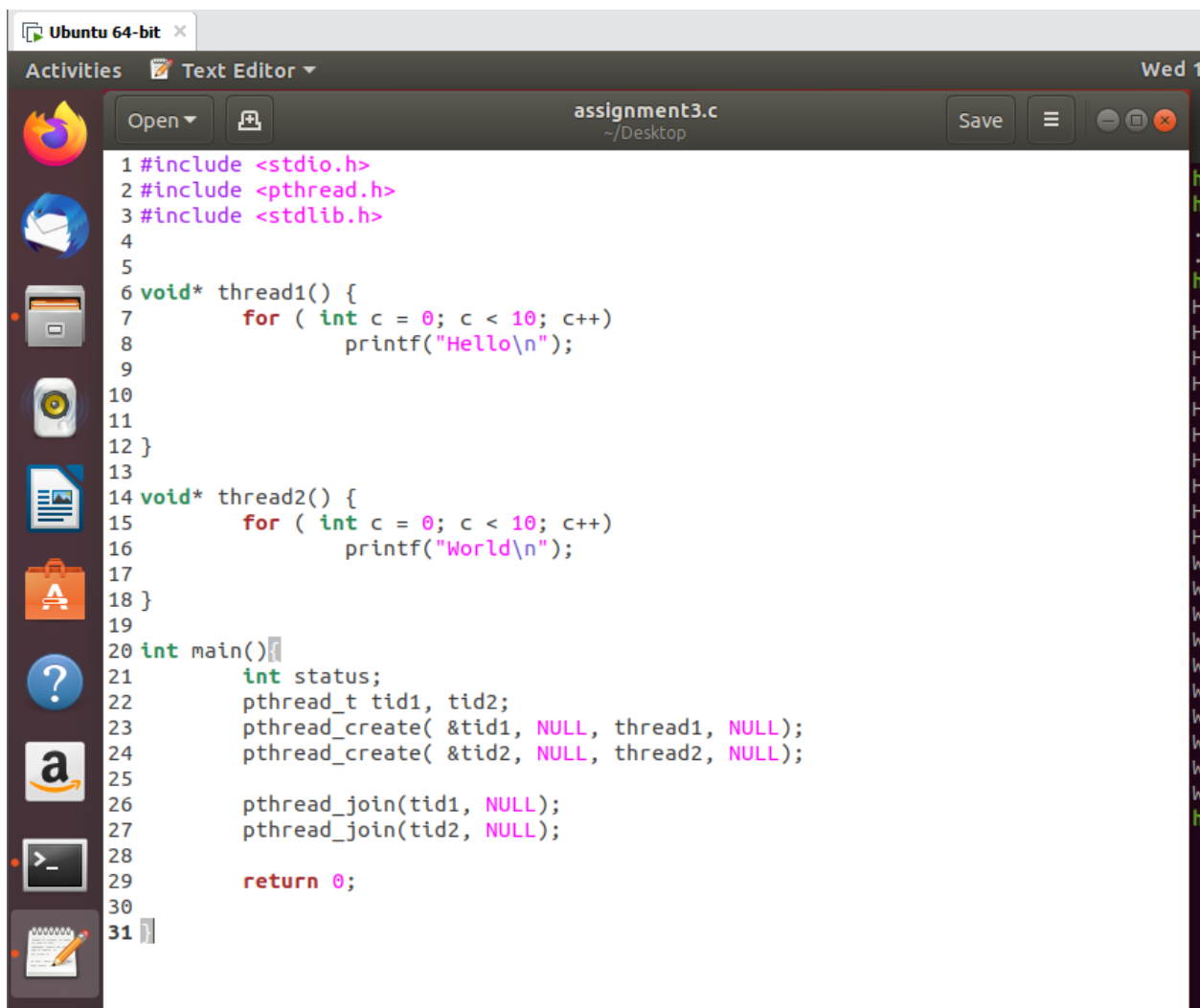
SECTION: B

Assignment: 03

SUBJECT: Operating System.

Question #01

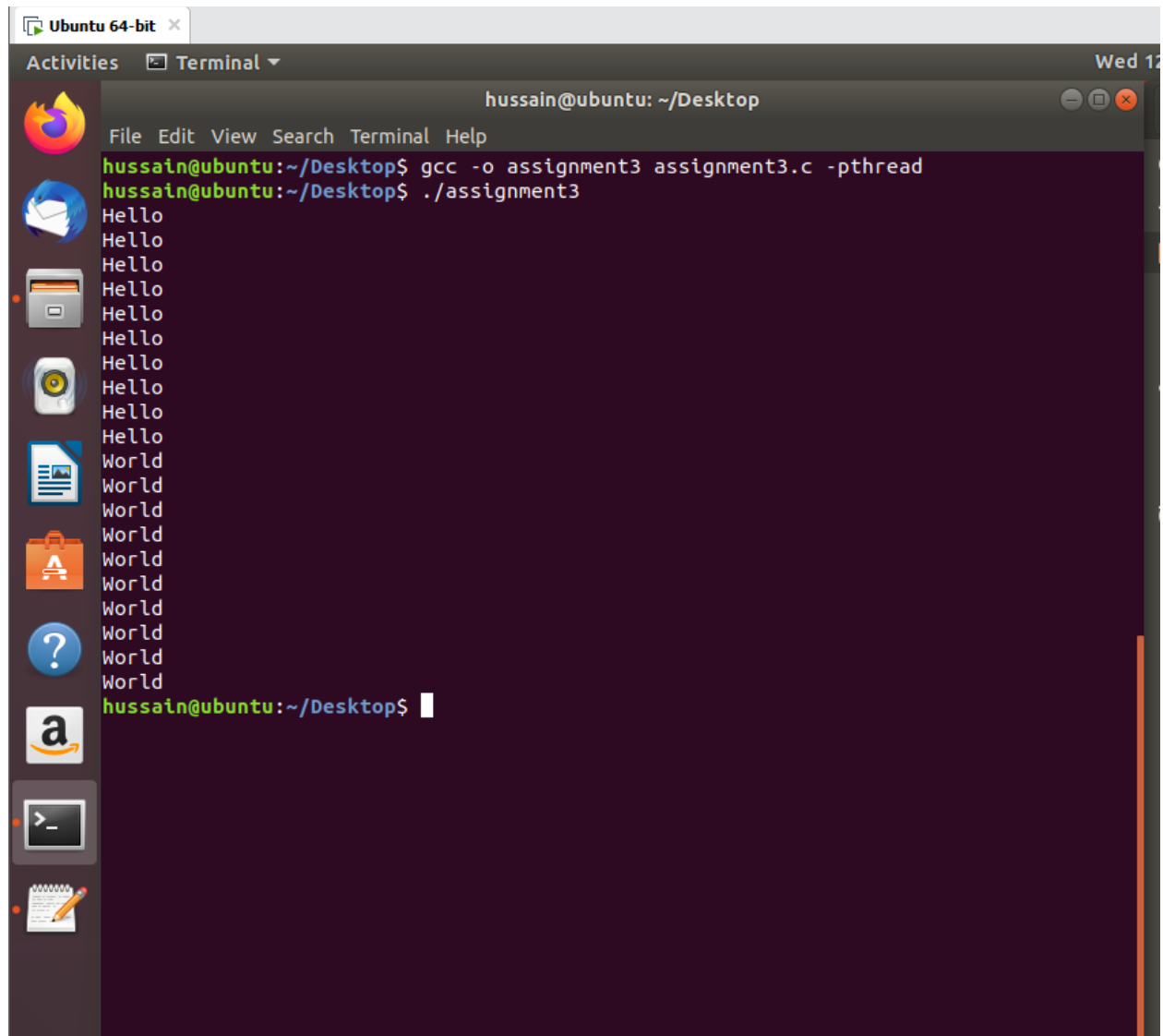
(a) Write the following program and observe and explain the output.



The screenshot shows a text editor window titled 'assignment3.c' on a desktop. The code is a C program that creates two threads, thread1 and thread2, and joins them back to the main thread. Thread1 prints 'Hello\n' 10 times, and thread2 prints 'World\n' 10 times. The main thread waits for both child threads to finish before returning 0.

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

(c) Output of above program:



```
hussain@ubuntu: ~/Desktop
File Edit View Search Terminal Help
hussain@ubuntu:~/Desktop$ gcc -o assignment3 assignment3.c -pthread
hussain@ubuntu:~/Desktop$ ./assignment3
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
World
World
World
World
World
World
World
World
World
World
World
hussain@ubuntu:~/Desktop$
```

Explanation:

As we see clearly that the thread1 prints “Hello” 10 times and thread2 print “World” 10 times, where both function are call from the main to the threads. The main process also waits for the thread to finish

pthread_create: means thread is created

pthread_join : means process waits for thread to finish .

(b) Modify the program to create four threads using the same two functions (thread1 and thread2).

A screenshot of a Linux desktop environment. The top panel shows the 'Activities' button, a 'Text Editor' window title, and the system clock 'Wed 13:06'. The text editor window displays a C program named 'assignmentb.c' located at '~/Desktop'. The code defines four threads: thread1 and thread2 both print 'Hello\n', while thread3 and thread4 both print 'World\n'. The main function creates these four threads using pthread_create and joins them using pthread_join. The code is as follows:

```
1 #include <stdio.h>
2 #include <pthread.h>
3 #include <stdlib.h>
4 void* thread1() {
5     for ( int c = 0; c < 10; c++)
6         printf("Hello\n");
7 }
8
9 void* thread2() {
10    for ( int c = 0; c < 10; c++)
11        printf("Hello\n");
12 }
13 void* thread3() {
14    for ( int c = 0; c < 10; c++)
15        printf("World\n");
16 }
17 void* thread4() {
18    for ( int c = 0; c < 10; c++)
19        printf("World\n");
20 }
21
22 int main(){
23     int status;
24     pthread_t tid1, tid2, tid3, tid4;
25     pthread_create( &tid1, NULL, thread1, NULL);
26     pthread_create( &tid2, NULL, thread2, NULL);
27     pthread_create( &tid3, NULL, thread3, NULL);
28     pthread_create( &tid4, NULL, thread4, NULL);
29
30     pthread_join(tid1, NULL);
31     pthread_join(tid2, NULL);
32     pthread_join(tid3, NULL);
33     pthread_join(tid4, NULL);
34
35     return 0;
36 }
37 }
```

(c) Output of above program:

[illegible]

Explanation:

As we see clearly that the thread1 prints “Hello” 10 times ,

thread2 print “World” 10 times,

thread3 prints "Hello" 10 times

and thread4 print “World” 10 times

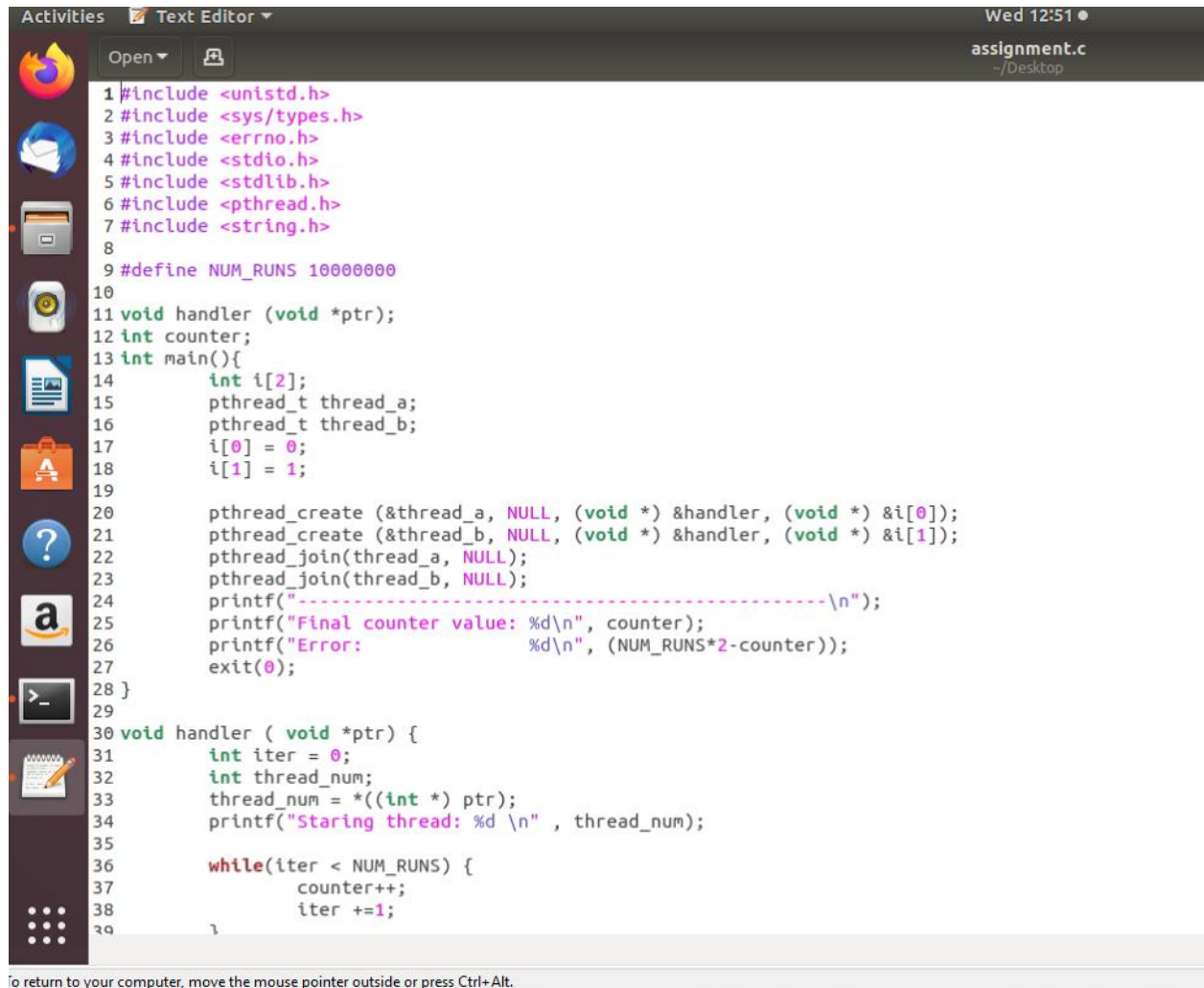
so, "hello" prints 20 times and "world" also print 20 times.

pthread_create: means thread is created

pthread_join : means process waits for thread to finish .

Question #02

(a) Compile and execute the program.

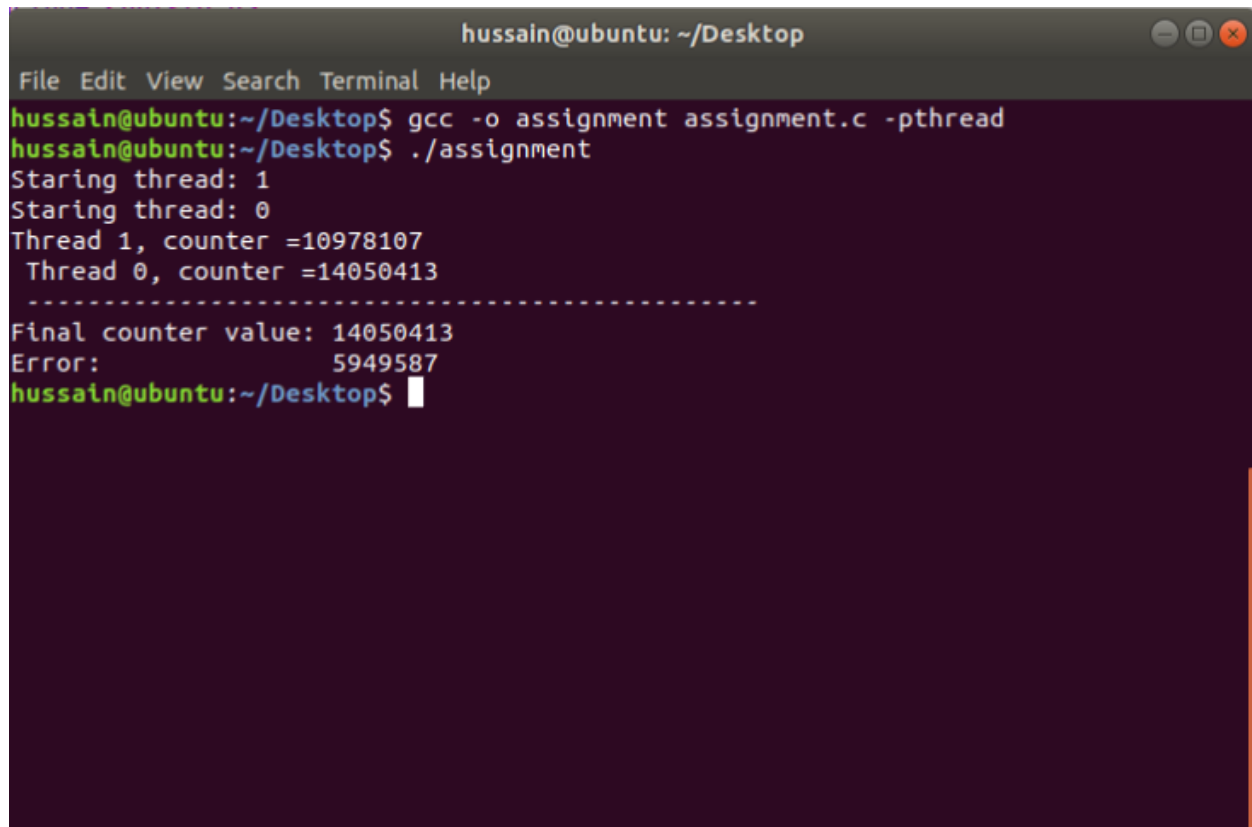


```
Activities Text Editor Wed 12:51
assignment.c ~/Desktop

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

To return to your computer, move the mouse pointer outside or press Ctrl+Alt.

Output of above program:



```
hussain@ubuntu: ~/Desktop
File Edit View Search Terminal Help
hussain@ubuntu:~/Desktop$ gcc -o assignment assignment.c -pthread
hussain@ubuntu:~/Desktop$ ./assignment
Starting thread: 1
Starting thread: 0
Thread 1, counter =10978107
Thread 0, counter =14050413
-----
Final counter value: 14050413
Error: 5949587
hussain@ubuntu:~/Desktop$
```

(b) Answer the following questions:

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

The value of counter variable at the end should be = 10978107

ii. What is the value you get?

The final value of the counter I get is: 14050413

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

The value of error is = 5949587 . if error is large then it takes time to run .

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

There are almost less than a second time take the program to run on my system.

