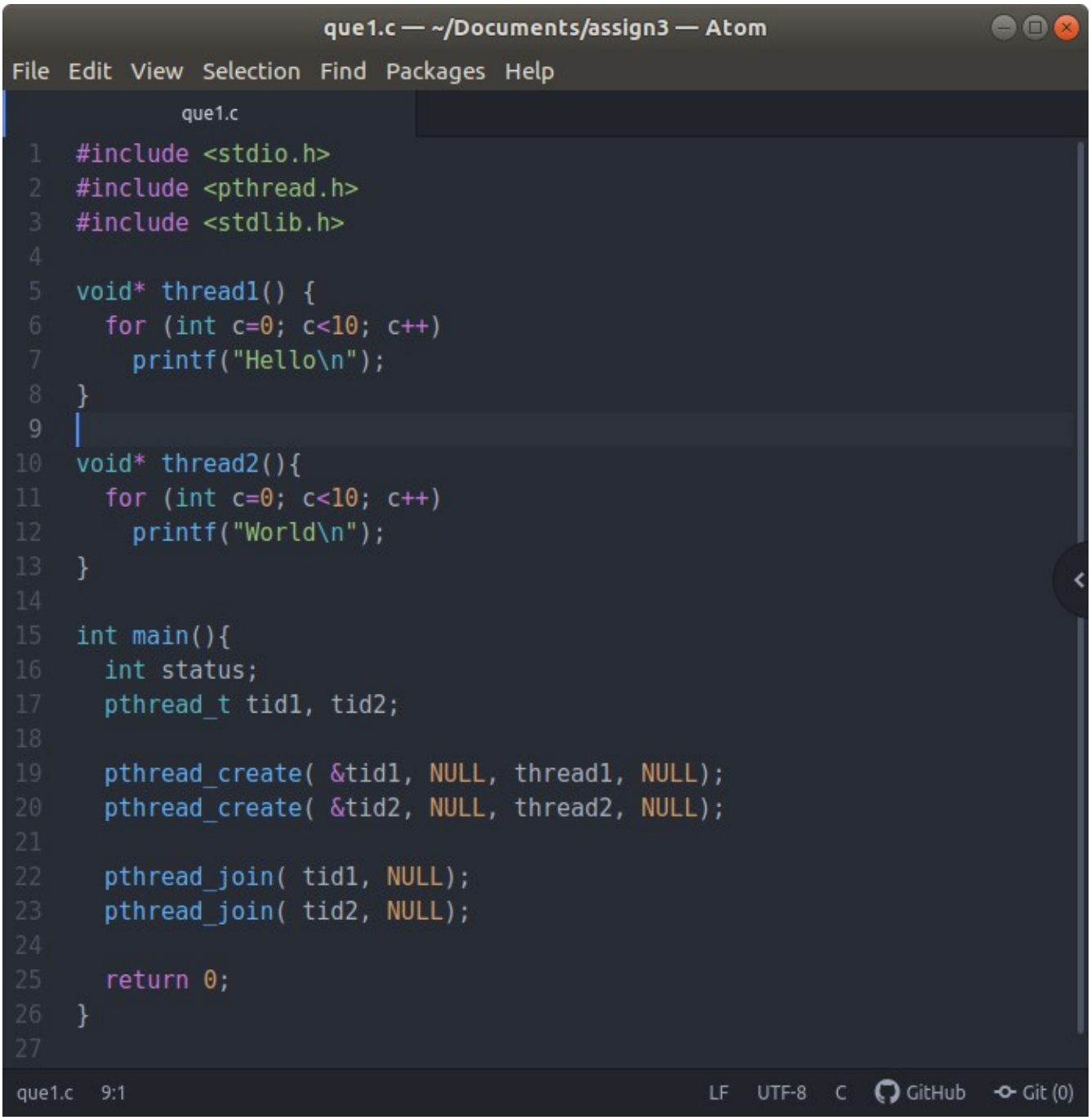
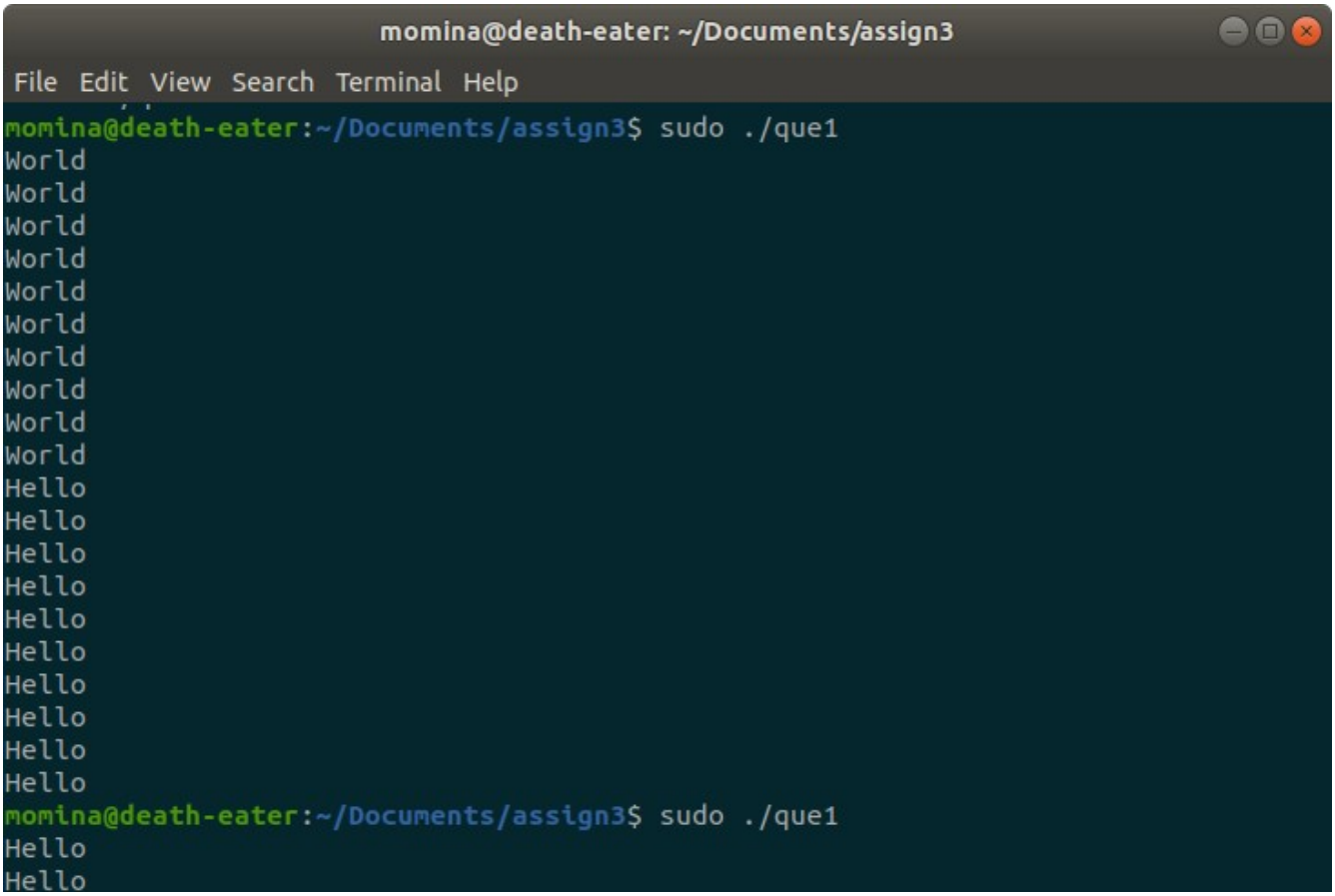
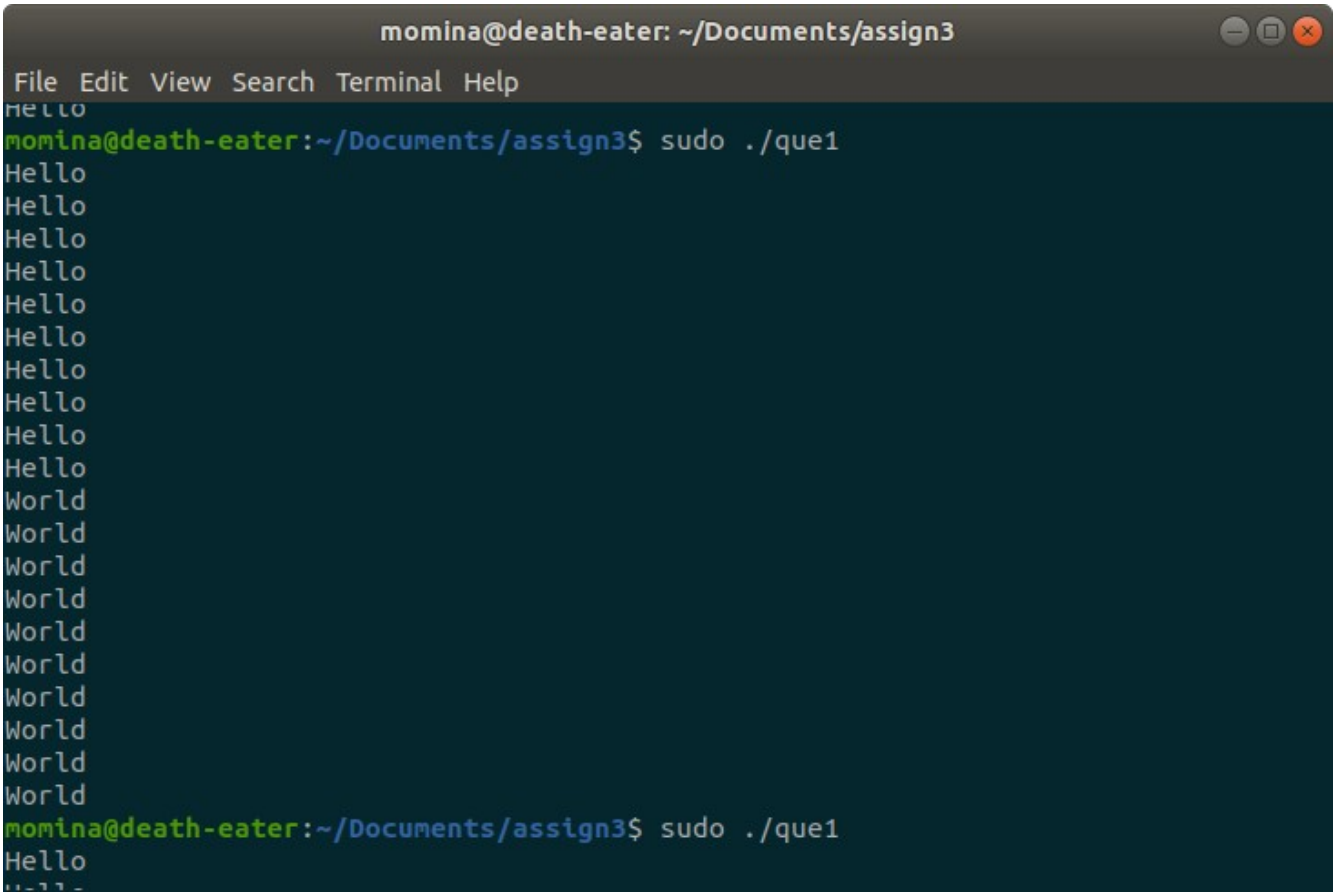
OPERATING SYSTEMS

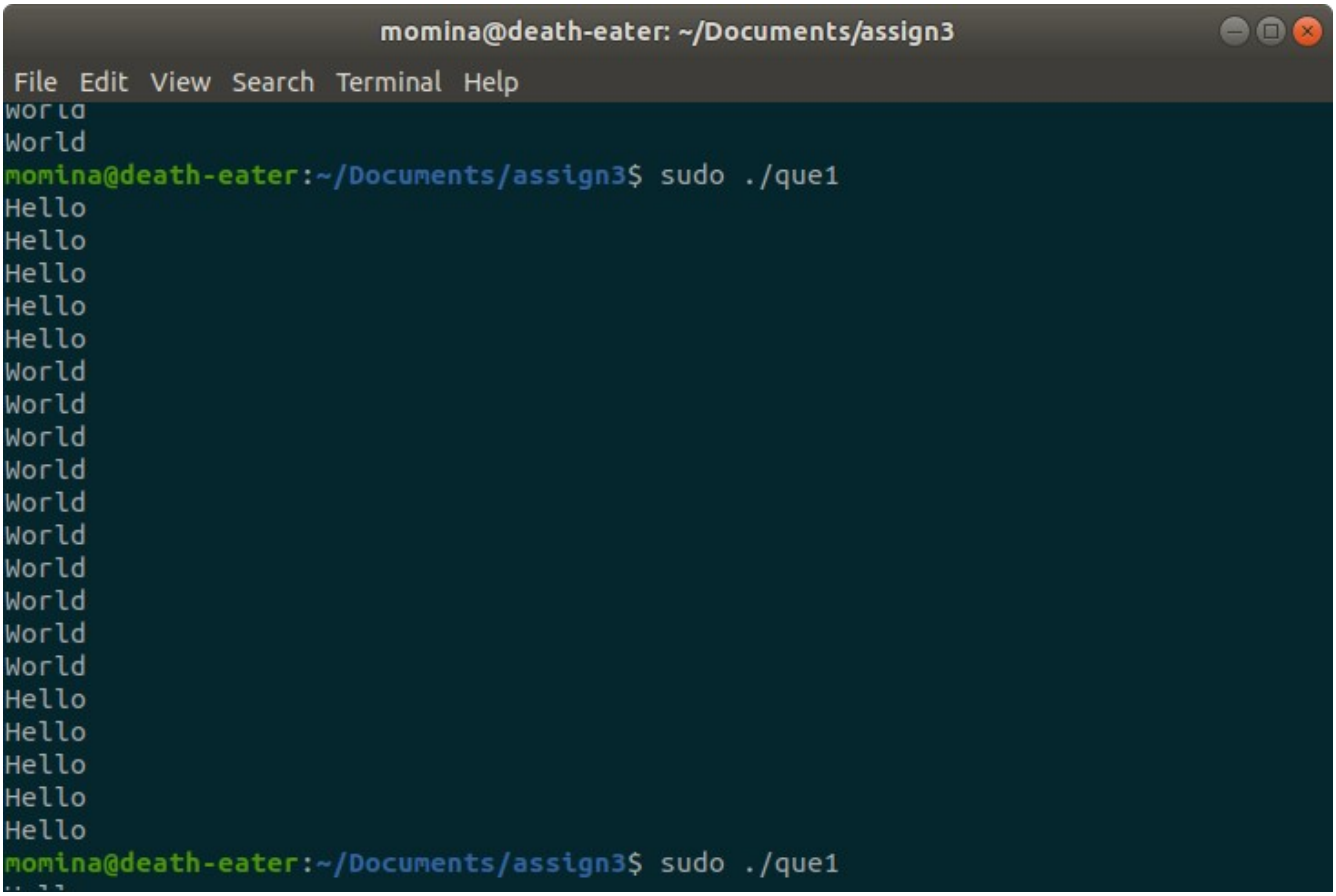
Momina Atif Dar
P18-0030
Section: B

Question 1:




momina@death-eater: ~/Documents/assign3
File Edit View Search Terminal Help
momina@death-eater:~/Documents/assign3$ sudo ./que1
World
World
World
World
World
World
World
World
World
World
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
momina@death-eater:~/Documents/assign3$ sudo ./que1
Hello
Hello


momina@death-eater: ~/Documents/assign3
File Edit View Search Terminal Help
momina@death-eater:~/Documents/assign3$ sudo ./que1
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
World
World
World
World
World
World
World
World
World
World
momina@death-eater:~/Documents/assign3$ sudo ./que1
Hello



Explanation:

Sometimes ten ‘Hello’ is printed first which means Thread 1 is executing first without occurrence of race condition/preemption and sometimes ten ‘World’ is printed first which means Thread 2 is executing first without any preemption. Sometimes exact opposite is happening with Thread 2 completely executing first, followed by complete execution of Thread 1.
Sometimes Thread 1 executes for less than ten times and is preempted, Thread 2 is given turn and it starts executing. Thread 2 completes its execution and Thread 1 resumes its execution from where it left.
It can also happen that in the above paragraph/example, Thread 2 doesn’t get to complete its execution and might be preempted. In that case Thread 1 will resume its execution. Preemption can occur at any point in the loops above, no matter what iteration of loop is running.

Modified Version Q1:

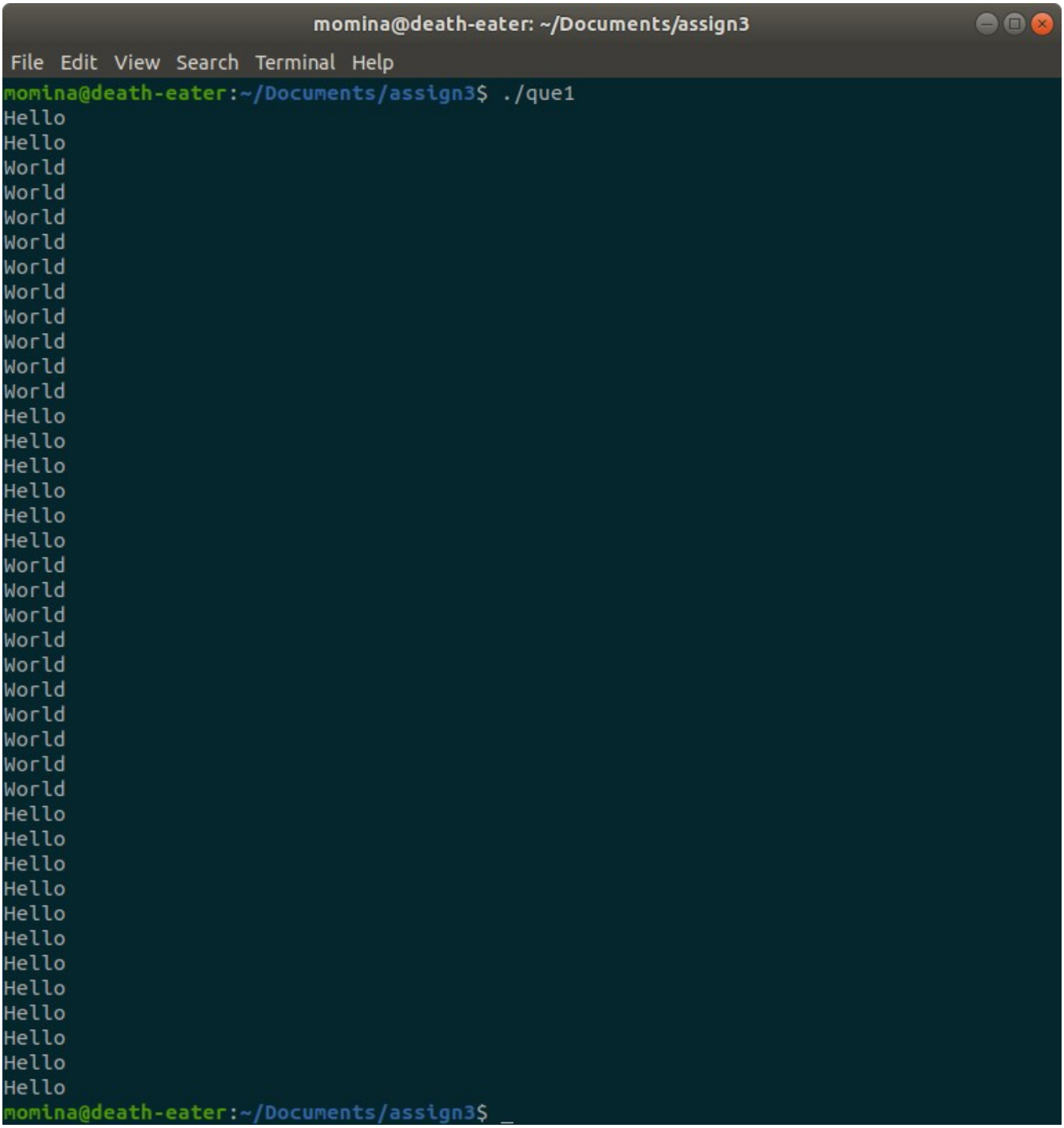
que1.c — ~/Documents/assign3 — Atom

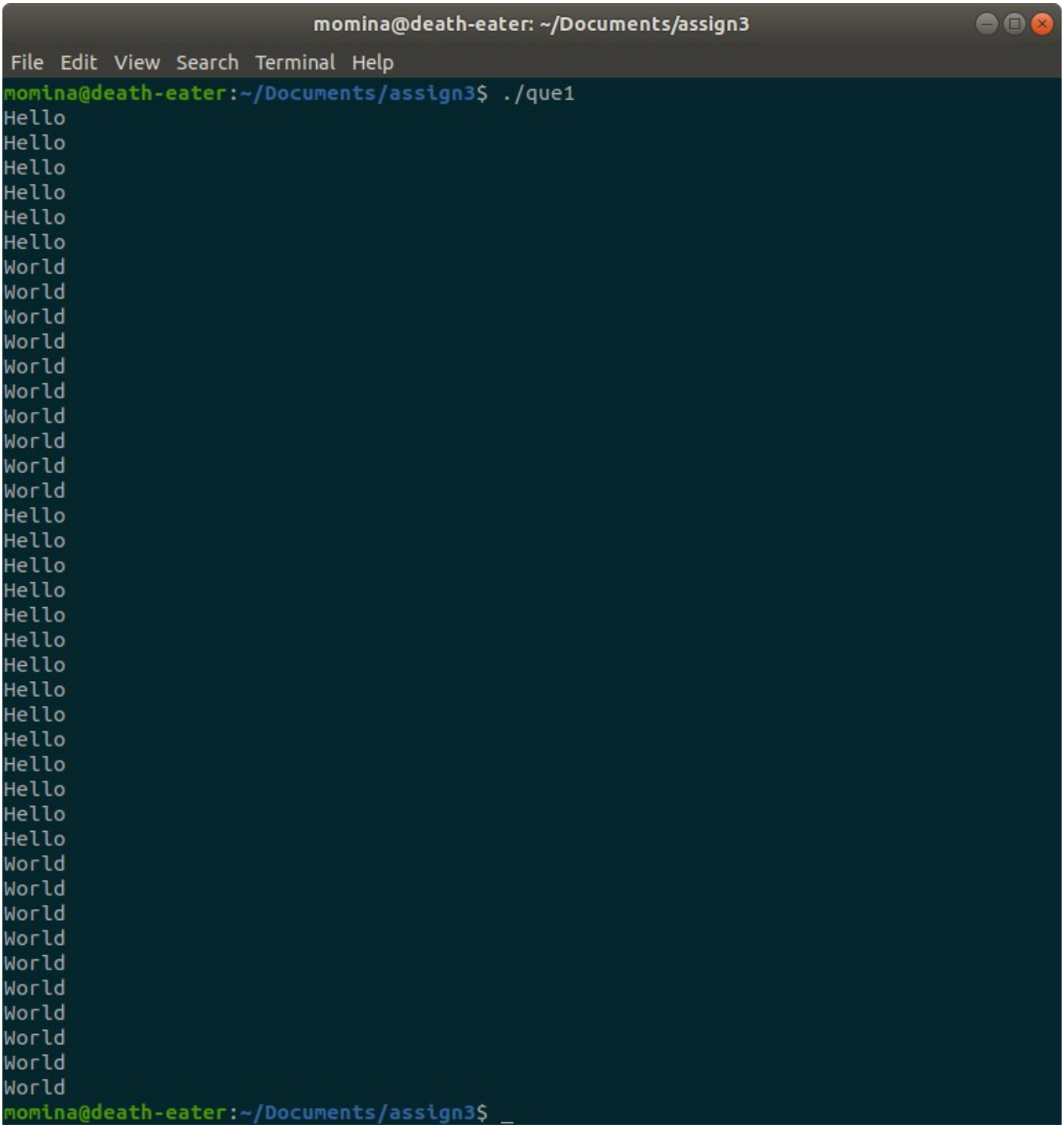
File Edit View Selection Find Packages Help

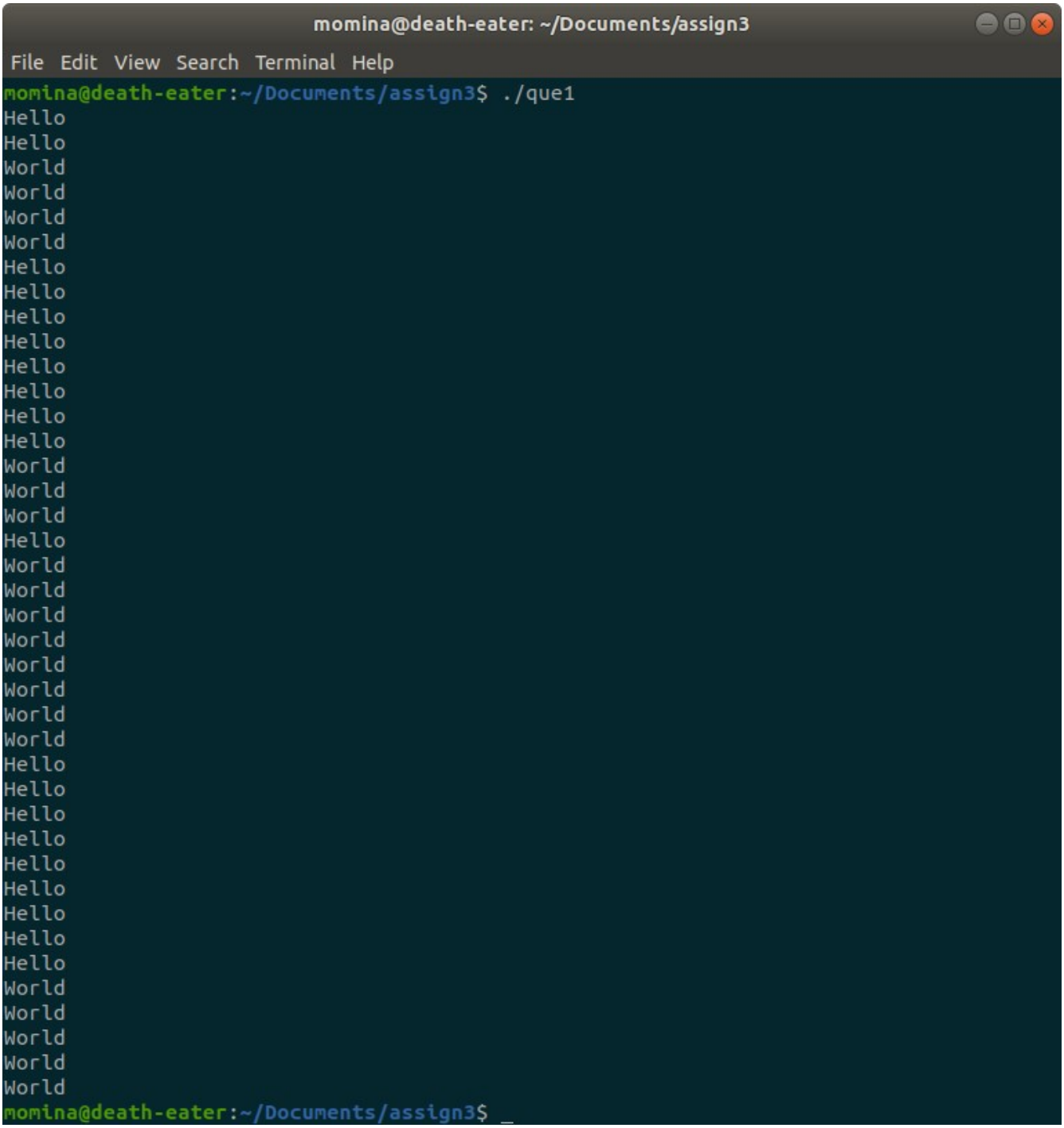
que1.c

```c
#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>

void* thread1() {
  for (int c=0; c<10; c++)
    printf("Hello\n");
}

void* thread2(){
  for (int c=0; c<10; c++)
    printf("World\n");
}

int main(){
  int status;
  pthread_t tid1, tid2, tid3, tid4;

  pthread_create( &tid1, NULL, thread1, NULL);
  pthread_create( &tid2, NULL, thread2, NULL);
  pthread_create( &tid3, NULL, thread1, NULL); //
  pthread_create( &tid4, NULL, thread2, NULL); //

  pthread_join( tid1, NULL);
  pthread_join( tid2, NULL);
  pthread_join( tid3, NULL); //
  pthread_join( tid4, NULL); //

  return 0;
}
```

~/Documents/assign3/que1.c 27:32 LF UTF-8 C GitHub Git (0)


momina@death-eater: ~/Documents/assign3
File Edit View Search Terminal Help
momina@death-eater:~/Documents/assign3$ ./que1
Hello
Hello
World
World
World
World
World
World
World
World
World
World
Hello
Hello
Hello
Hello
Hello
Hello
World
World
World
World
World
World
World
World
World
World
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
momina@death-eater:~/Documents/assign3$ _


momina@death-eater: ~/Documents/assign3
File Edit View Search Terminal Help
momina@death-eater:~/Documents/assign3$ ./que1
Hello
Hello
Hello
Hello
Hello
Hello
Hello
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
World
World
World
World
World
World
World
World
World
World
momina@death-eater:~/Documents/assign3$ _



Explanation:

Almost same explanation as above but since two threads will print ‘Hello’ and two will print ‘World’ I cannot clearly say anything about the order of execution. But I can assume that in the last output image, Thread 1 executes, preemption occurs and Thread 2 or Thread 4 has been given the turn to execute, or maybe Thread 2 and Thread 4 are running one after another (printing two ‘World’ each). Then again preemption occurs and turn is given to Thread 1 or Thread 3 or maybe both after one another. This goes on.

Question 2:

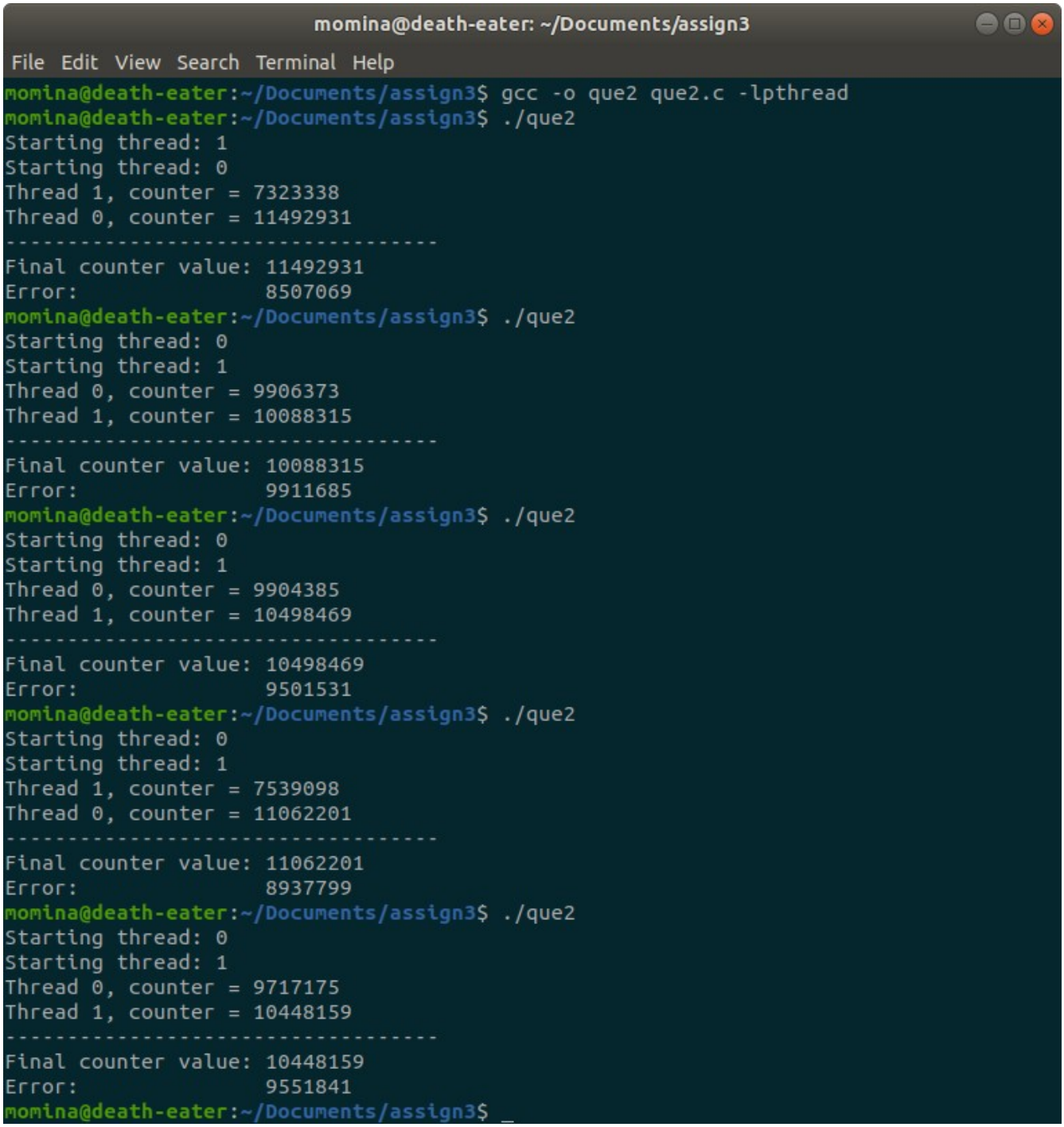
que2.c — ~/Documents/assign3 — Atom

File Edit View Selection Find Packages Help

que1.c | que2.c

```c
#include <unistd.h>
#include <sys/types.h>
#include <errno.h>
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <string.h>

#define NUM_RUNS 10000000

void handler(void *ptr);

int counter;

int main(){
  int i[2];
  pthread_t thread_a;
  pthread_t thread_b;

  i[0] = 0;
  i[1] = 1;

  pthread_create(&thread_a, NULL, (void *) &handler, (void *) &i[0]);
  pthread_create(&thread_b, NULL, (void *) &handler, (void *) &i[1]);

  pthread_join(thread_a, NULL);
  pthread_join(thread_b, NULL);

  printf("-----------------------------------\n");
  printf("Final counter value: %d\n", counter);
  printf("Error:               %d\n", (NUM_RUNS*2-counter));

  exit(0);
}

void handler(void *ptr){
  int iter = 0;
  int thread_num;

  thread_num = *((int *) ptr);

  printf("Starting thread: %d \n", thread_num);

  while(iter < NUM_RUNS){
    counter++;      //20000000
    iter += 1;
  }

  printf("Thread %d, counter = %d \n", thread_num, counter);

  pthread_exit(0);

```

que2.c 40:31 LF UTF-8 C GitHub Git (0)


momina@death-eater: ~/Documents/assign3
File Edit View Search Terminal Help
momina@death-eater:~/Documents/assign3$ gcc -o que2 que2.c -lpthread
momina@death-eater:~/Documents/assign3$ ./que2
Starting thread: 1
Starting thread: 0
Thread 1, counter = 7323338
Thread 0, counter = 11492931

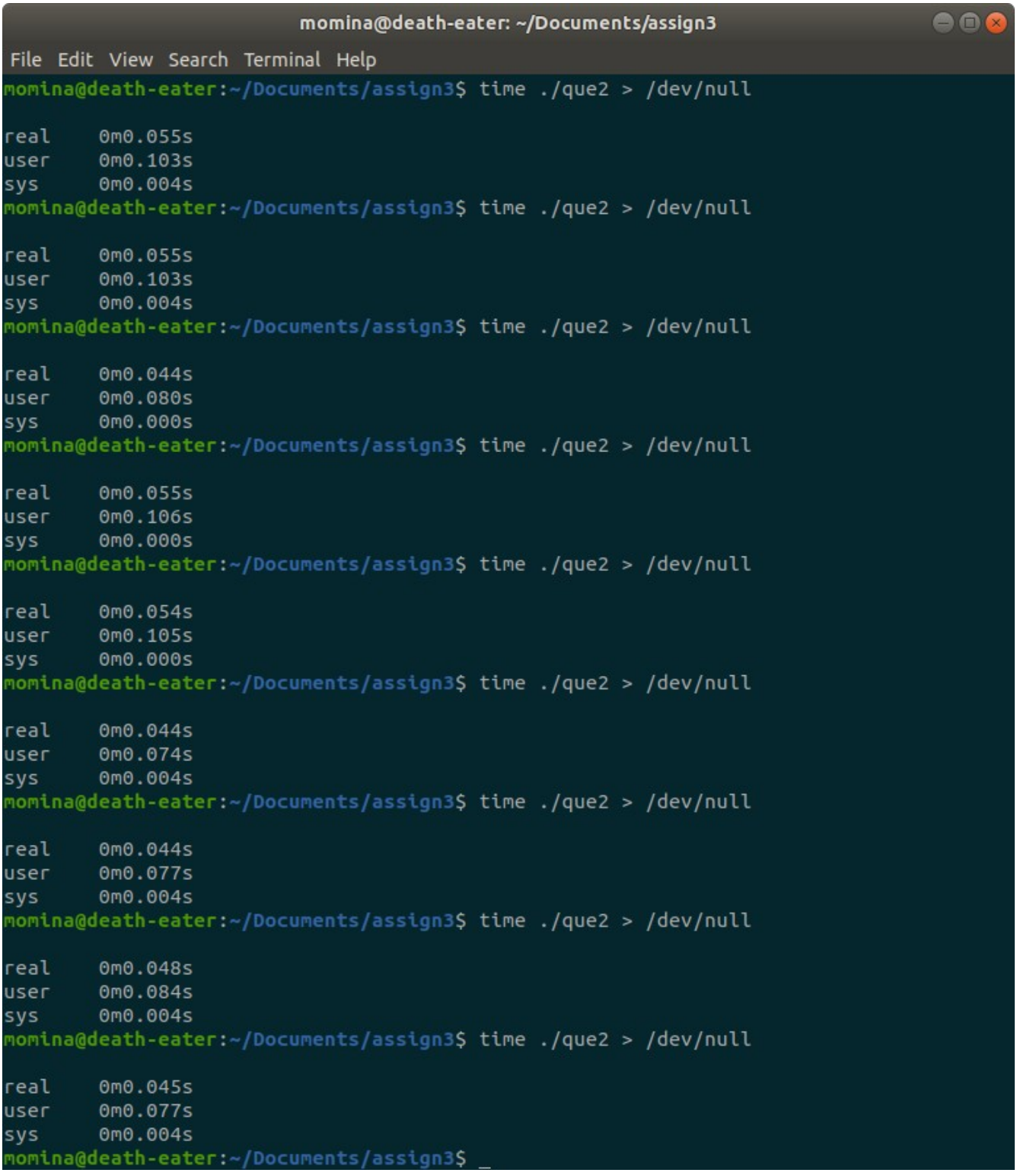
Final counter value: 11492931
Error: 8507069
momina@death-eater:~/Documents/assign3$ ./que2
Starting thread: 0
Starting thread: 1
Thread 0, counter = 9906373
Thread 1, counter = 10088315

Final counter value: 10088315
Error: 9911685
momina@death-eater:~/Documents/assign3$ ./que2
Starting thread: 0
Starting thread: 1
Thread 0, counter = 9904385
Thread 1, counter = 10498469

Final counter value: 10498469
Error: 9501531
momina@death-eater:~/Documents/assign3$ ./que2
Starting thread: 0
Starting thread: 1
Thread 1, counter = 7539098
Thread 0, counter = 11062201

Final counter value: 11062201
Error: 8937799
momina@death-eater:~/Documents/assign3$ ./que2
Starting thread: 0
Starting thread: 1
Thread 0, counter = 9717175
Thread 1, counter = 10448159

Final counter value: 10448159
Error: 9551841
momina@death-eater:~/Documents/assign3$ _


momina@death-eater: ~/Documents/assign3
File Edit View Search Terminal Help
momina@death-eater:~/Documents/assign3$ time ./que2 > /dev/null
real 0m0.055s
user 0m0.103s
sys 0m0.004s
momina@death-eater:~/Documents/assign3$ time ./que2 > /dev/null
real 0m0.055s
user 0m0.103s
sys 0m0.004s
momina@death-eater:~/Documents/assign3$ time ./que2 > /dev/null
real 0m0.044s
user 0m0.080s
sys 0m0.000s
momina@death-eater:~/Documents/assign3$ time ./que2 > /dev/null
real 0m0.055s
user 0m0.106s
sys 0m0.000s
momina@death-eater:~/Documents/assign3$ time ./que2 > /dev/null
real 0m0.054s
user 0m0.105s
sys 0m0.000s
momina@death-eater:~/Documents/assign3$ time ./que2 > /dev/null
real 0m0.044s
user 0m0.074s
sys 0m0.004s
momina@death-eater:~/Documents/assign3$ time ./que2 > /dev/null
real 0m0.044s
user 0m0.077s
sys 0m0.004s
momina@death-eater:~/Documents/assign3$ time ./que2 > /dev/null
real 0m0.048s
user 0m0.084s
sys 0m0.004s
momina@death-eater:~/Documents/assign3$ time ./que2 > /dev/null
real 0m0.045s
user 0m0.077s
sys 0m0.004s
momina@death-eater:~/Documents/assign3$ _

Questions:

i. 20000000

ii. 10448159

iii. 9551841. Sometimes it increases to approximately 1400000 and sometimes it decreases to approximately 400000 (by taking the difference between errors).

iv. By taking average, 0.089s.