UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE
KATHMANDU

[Workshop Report-3]

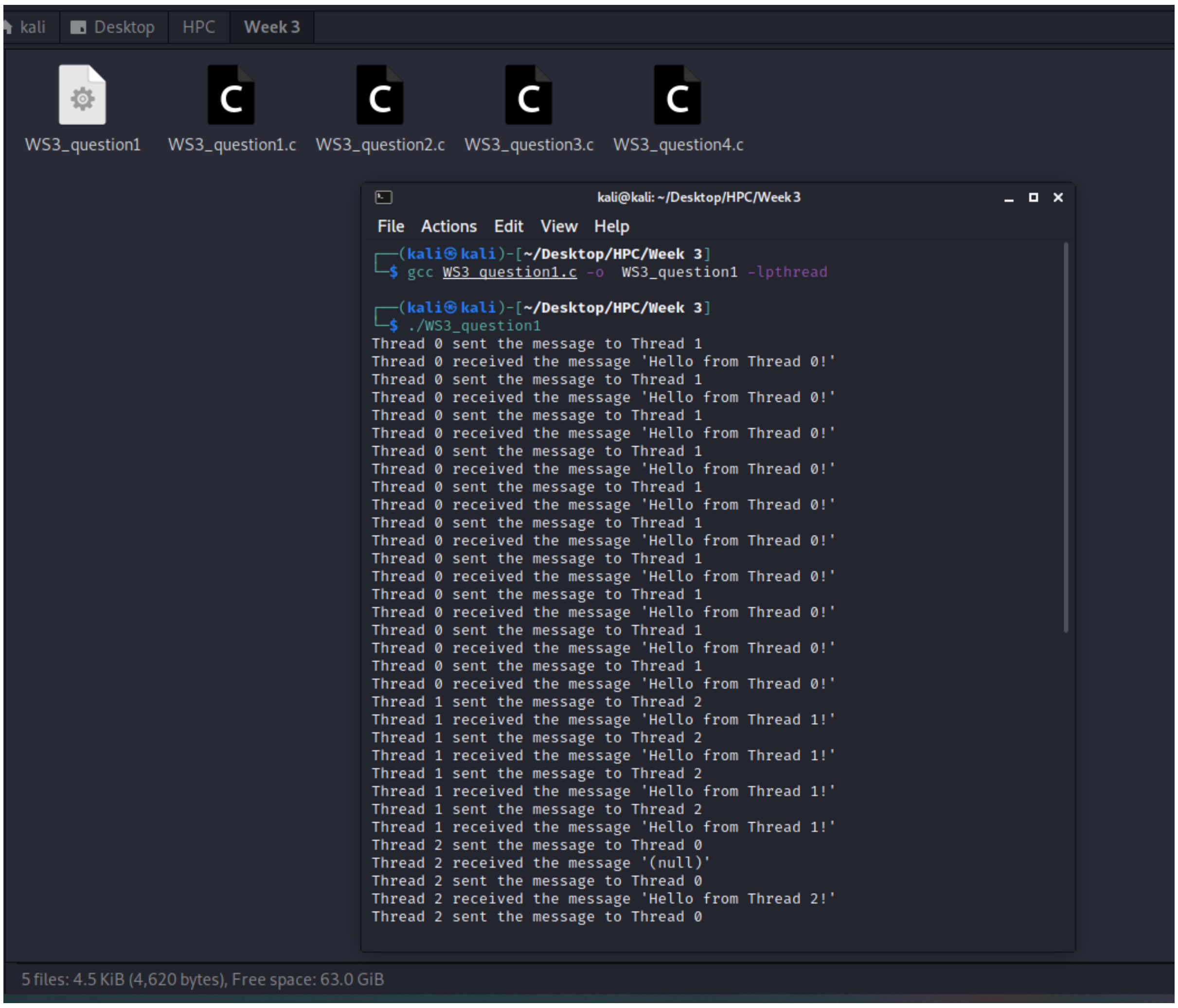
Student Id	: NP03A190051
Student Name	: Rishab Sharma
Section	: L5CG7
Lecturer	: Jnaneshwor Bohara
Module Leader	: Hiran Patel
Submitted on	: 1st October, 2021

1. 1. The following program demonstrates 3 thread sending string messages to each other, using a global array. The messages are sent meant to be sent in the following order:
   a. Thread 0 sends Thread 1 a message
   b. Thread 1 receives the message
   c. Thread 1 sends Thread 2 a message
   d. Thread 2 receives the message
   e. Thread 2 sends Thread 0 a message
   f. Thread 0 receives the message
   g. This then repeats from (a) 10 times

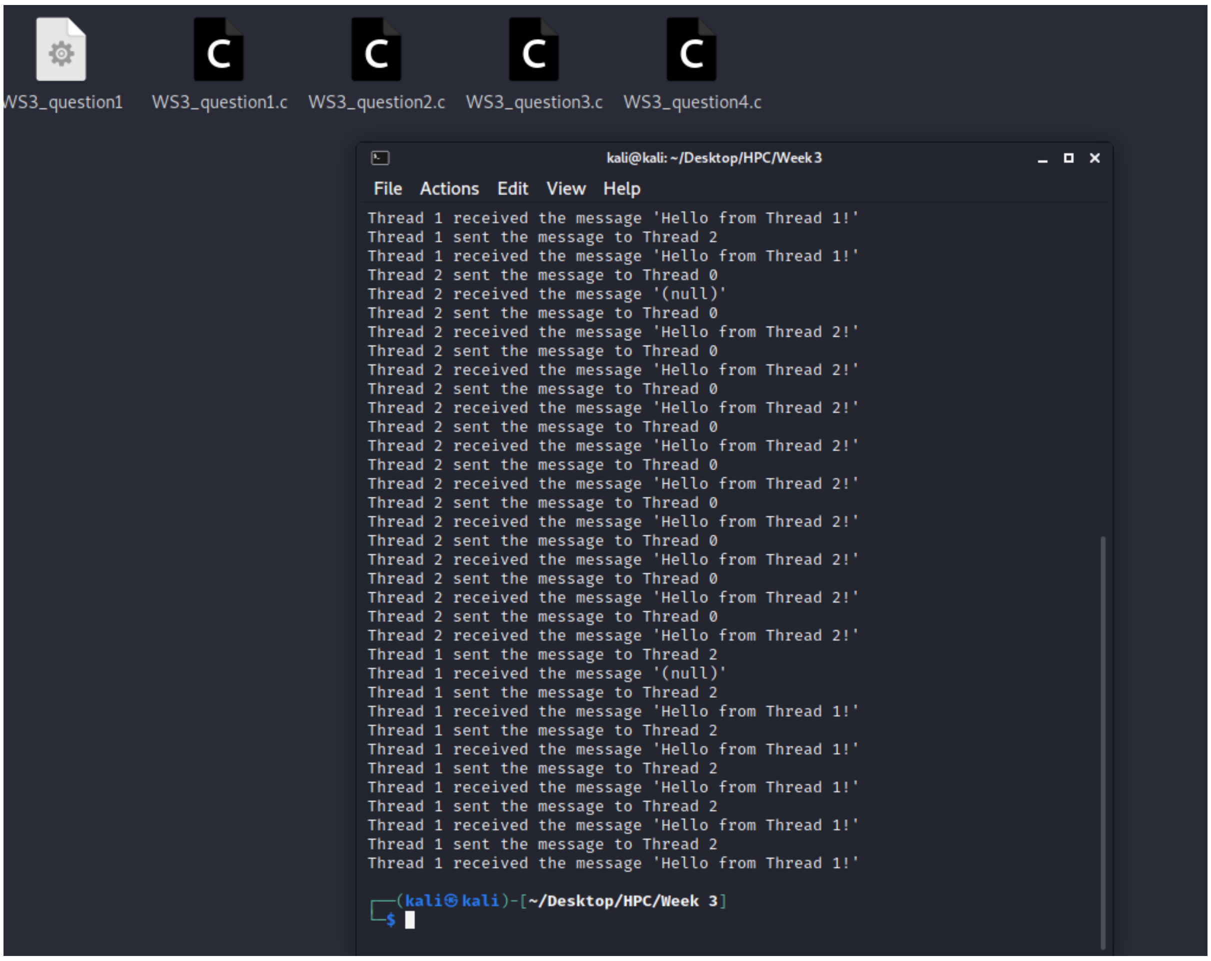
```
=> #include <stdio.h>
   #include <stdlib.h>
   #include <string.h>
   #include <pthread.h>
   #include <unistd.h>


char *messages[3] = {NULL, NULL, NULL};
void *messenger(void *p)
{
   long tid = (long)p;
   char tmpbuf[100];
   for(int i=0; i<10; i++)
 {
      /* Sending a message */
      long int dest = (tid + 1) % 3;
      sprintf(tmpbuf,"Hello from Thread %ld!", tid);
      char *msg = strdup(tmpbuf);
      messages[dest] = msg;
      printf("Thread %ld sent the message to Thread %ld\n",tid, dest);
      /* Receiving a message */
      printf("Thread %ld received the message '%s'\n",tid, messages[dest]);
      free(messages[tid]);
      messages[tid] = NULL;
```

```
    }
}

void main() {
    pthread_t thrID1, thrID2, thrID3;
    pthread_create(&thrID1, NULL, messenger, (void *)0);
    pthread_create(&thrID2, NULL, messenger, (void *)1);
    pthread_create(&thrID3, NULL, messenger, (void *)2);
    pthread_join(thrID1, NULL);
    pthread_join(thrID2, NULL);
    pthread_join(thrID3, NULL);
}
```

OUTPUT: -


kali
Desktop
HPC
Week 3
WS3_question1
WS3_question1.c
WS3_question2.c
WS3_question3.c
WS3_question4.c
kali@kali: ~/Desktop/HPC/Week 3
File Actions Edit View Help
5 files: 4.5 KiB (4,620 bytes), Free space: 63.0 GiB

```
┌──(kali㉿kali)-[~/Desktop/HPC/Week 3]
└─$ gcc WS3_question1.c -o  WS3_question1 -lpthread

┌──(kali㉿kali)-[~/Desktop/HPC/Week 3]
└─$ ./WS3_question1
Thread 0 sent the message to Thread 1
Thread 0 received the message 'Hello from Thread 0!'
Thread 0 sent the message to Thread 1
Thread 0 received the message 'Hello from Thread 0!'
Thread 0 sent the message to Thread 1
Thread 0 received the message 'Hello from Thread 0!'
Thread 0 sent the message to Thread 1
Thread 0 received the message 'Hello from Thread 0!'
Thread 0 sent the message to Thread 1
Thread 0 received the message 'Hello from Thread 0!'
Thread 0 sent the message to Thread 1
Thread 0 received the message 'Hello from Thread 0!'
Thread 0 sent the message to Thread 1
Thread 0 received the message 'Hello from Thread 0!'
Thread 0 sent the message to Thread 1
Thread 0 received the message 'Hello from Thread 0!'
Thread 0 sent the message to Thread 1
Thread 0 received the message 'Hello from Thread 0!'
Thread 0 sent the message to Thread 1
Thread 0 received the message 'Hello from Thread 0!'
Thread 1 sent the message to Thread 2
Thread 1 received the message 'Hello from Thread 1!'
Thread 1 sent the message to Thread 2
Thread 1 received the message 'Hello from Thread 1!'
Thread 1 sent the message to Thread 2
Thread 1 received the message 'Hello from Thread 1!'
Thread 1 sent the message to Thread 2
Thread 1 received the message 'Hello from Thread 1!'
Thread 2 sent the message to Thread 0
Thread 2 received the message '(null)'
Thread 2 sent the message to Thread 0
Thread 2 received the message 'Hello from Thread 2!'
Thread 2 sent the message to Thread 0
```



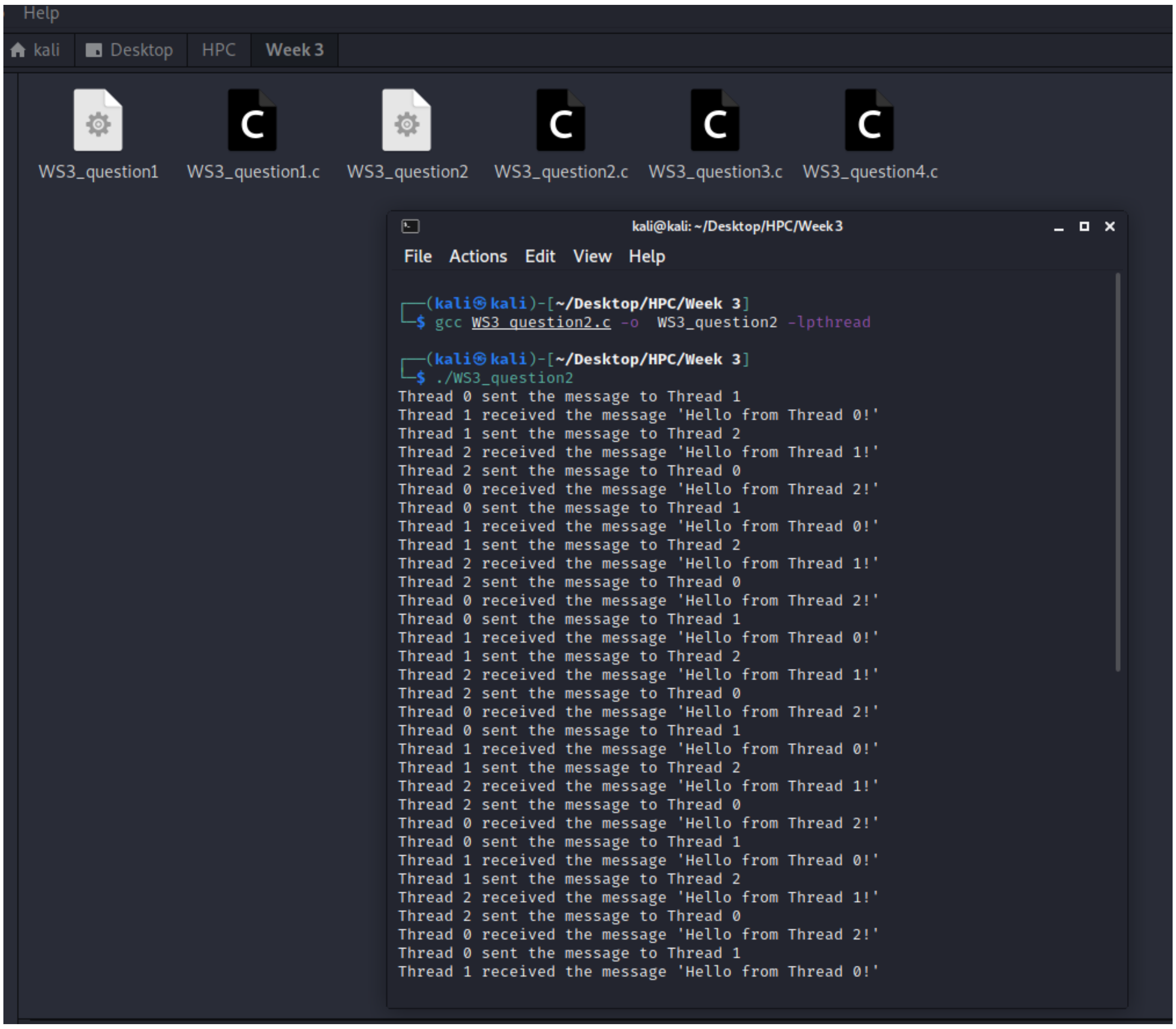
```
Thread 1 received the message 'Hello from Thread 1!'
Thread 1 sent the message to Thread 2
Thread 1 received the message 'Hello from Thread 1!'
Thread 2 sent the message to Thread 0
Thread 2 received the message '(null)'
Thread 2 sent the message to Thread 0
Thread 2 received the message 'Hello from Thread 2!'
Thread 2 sent the message to Thread 0
Thread 2 received the message 'Hello from Thread 2!'
Thread 2 sent the message to Thread 0
Thread 2 received the message 'Hello from Thread 2!'
Thread 2 sent the message to Thread 0
Thread 2 received the message 'Hello from Thread 2!'
Thread 2 sent the message to Thread 0
Thread 2 received the message 'Hello from Thread 2!'
Thread 2 sent the message to Thread 0
Thread 2 received the message 'Hello from Thread 2!'
Thread 2 sent the message to Thread 0
Thread 2 received the message 'Hello from Thread 2!'
Thread 2 sent the message to Thread 0
Thread 2 received the message 'Hello from Thread 2!'
Thread 2 sent the message to Thread 0
Thread 2 received the message 'Hello from Thread 2!'
Thread 1 sent the message to Thread 2
Thread 1 received the message '(null)'
Thread 1 sent the message to Thread 2
Thread 1 received the message 'Hello from Thread 1!'
Thread 1 sent the message to Thread 2
Thread 1 received the message 'Hello from Thread 1!'
Thread 1 sent the message to Thread 2
Thread 1 received the message 'Hello from Thread 1!'
Thread 1 sent the message to Thread 2
Thread 1 received the message 'Hello from Thread 1!'
Thread 1 sent the message to Thread 2
Thread 1 received the message 'Hello from Thread 1!'

┌──(kali㉿kali)-[~/Desktop/HPC/Week 3]
└─$ 
```

2. Using the technique of "busy-waiting" to correct the program, and establishing the correct order of messages.

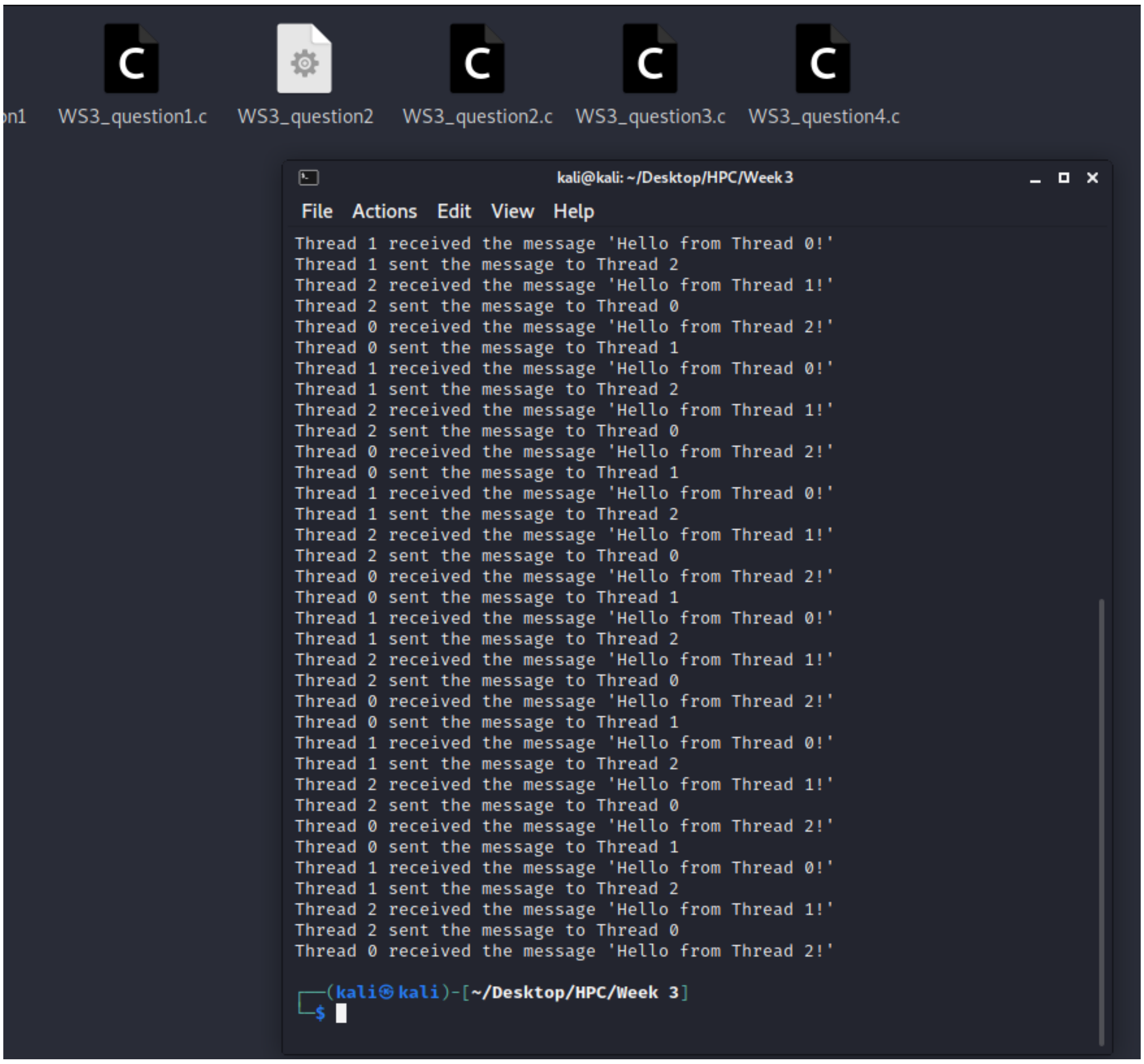
```
=> #include <stdio.h>
   #include <stdlib.h>
   #include <string.h>
   #include <pthread.h>
   #include <unistd.h>

char *messages[3] = {NULL, NULL, NULL};
int flag = 0;

void *messenger(void *p)
{
   long tid = (long)p;
```

```
    char tmpbuf[100];
        for(int i=0; i<10; i++)
        {
            while(flag!=tid);
            /* Sending a message */
            long int dest = (tid + 1) % 3;
            sprintf(tmpbuf,"Hello from Thread %ld!", tid);
            char *msg = strdup(tmpbuf);
            messages[dest] = msg;
            printf("Thread %ld sent the message to Thread %ld\n",tid, dest);


            /* Receiving a message */
            printf("Thread %ld received the message '%s'\n",dest, messages[dest]);
            free(messages[dest]);
            messages[dest] = NULL;
            flag = dest;
        }
    return NULL;
}


void main()
{
    pthread_t thrID1, thrID2, thrID3;
    pthread_create(&thrID1, NULL, messenger, (void *)0);
    pthread_create(&thrID2, NULL, messenger, (void *)1);
    pthread_create(&thrID3, NULL, messenger, (void *)2);
    pthread_join(thrID1, NULL);
    pthread_join(thrID2, NULL);
    pthread_join(thrID3, NULL);
}
```

OUTPUT: -



```
┌──(kali㉿kali)-[~/Desktop/HPC/Week 3]
└─$ gcc WS3_question2.c -o  WS3_question2 -lpthread

┌──(kali㉿kali)-[~/Desktop/HPC/Week 3]
└─$ ./WS3_question2
Thread 0 sent the message to Thread 1
Thread 1 received the message 'Hello from Thread 0!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 2 sent the message to Thread 0
Thread 0 received the message 'Hello from Thread 2!'
Thread 0 sent the message to Thread 1
Thread 1 received the message 'Hello from Thread 0!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 2 sent the message to Thread 0
Thread 0 received the message 'Hello from Thread 2!'
Thread 0 sent the message to Thread 1
Thread 1 received the message 'Hello from Thread 0!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 2 sent the message to Thread 0
Thread 0 received the message 'Hello from Thread 2!'
Thread 0 sent the message to Thread 1
Thread 1 received the message 'Hello from Thread 0!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 2 sent the message to Thread 0
Thread 0 received the message 'Hello from Thread 2!'
Thread 0 sent the message to Thread 1
Thread 1 received the message 'Hello from Thread 0!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 2 sent the message to Thread 0
Thread 0 received the message 'Hello from Thread 2!'
Thread 0 sent the message to Thread 1
Thread 1 received the message 'Hello from Thread 0!'
```



```
Thread 1 received the message 'Hello from Thread 0!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 2 sent the message to Thread 0
Thread 0 received the message 'Hello from Thread 2!'
Thread 0 sent the message to Thread 1
Thread 1 received the message 'Hello from Thread 0!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 2 sent the message to Thread 0
Thread 0 received the message 'Hello from Thread 2!'
Thread 0 sent the message to Thread 1
Thread 1 received the message 'Hello from Thread 0!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 2 sent the message to Thread 0
Thread 0 received the message 'Hello from Thread 2!'
Thread 0 sent the message to Thread 1
Thread 1 received the message 'Hello from Thread 0!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 2 sent the message to Thread 0
Thread 0 received the message 'Hello from Thread 2!'
Thread 0 sent the message to Thread 1
Thread 1 received the message 'Hello from Thread 0!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 2 sent the message to Thread 0
Thread 0 received the message 'Hello from Thread 2!'
Thread 0 sent the message to Thread 1
Thread 1 received the message 'Hello from Thread 0!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 2 sent the message to Thread 0
Thread 0 received the message 'Hello from Thread 2!'

┌──(kali㉿kali)-[~/Desktop/HPC/Week 3]
└─$ 
```

3. Use pthread “mutex” to correct the program in (1). You will need multiple mutexes.

=> #include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <pthread.h>

#include <unistd.h>

char *messages[3] = {NULL, NULL, NULL};

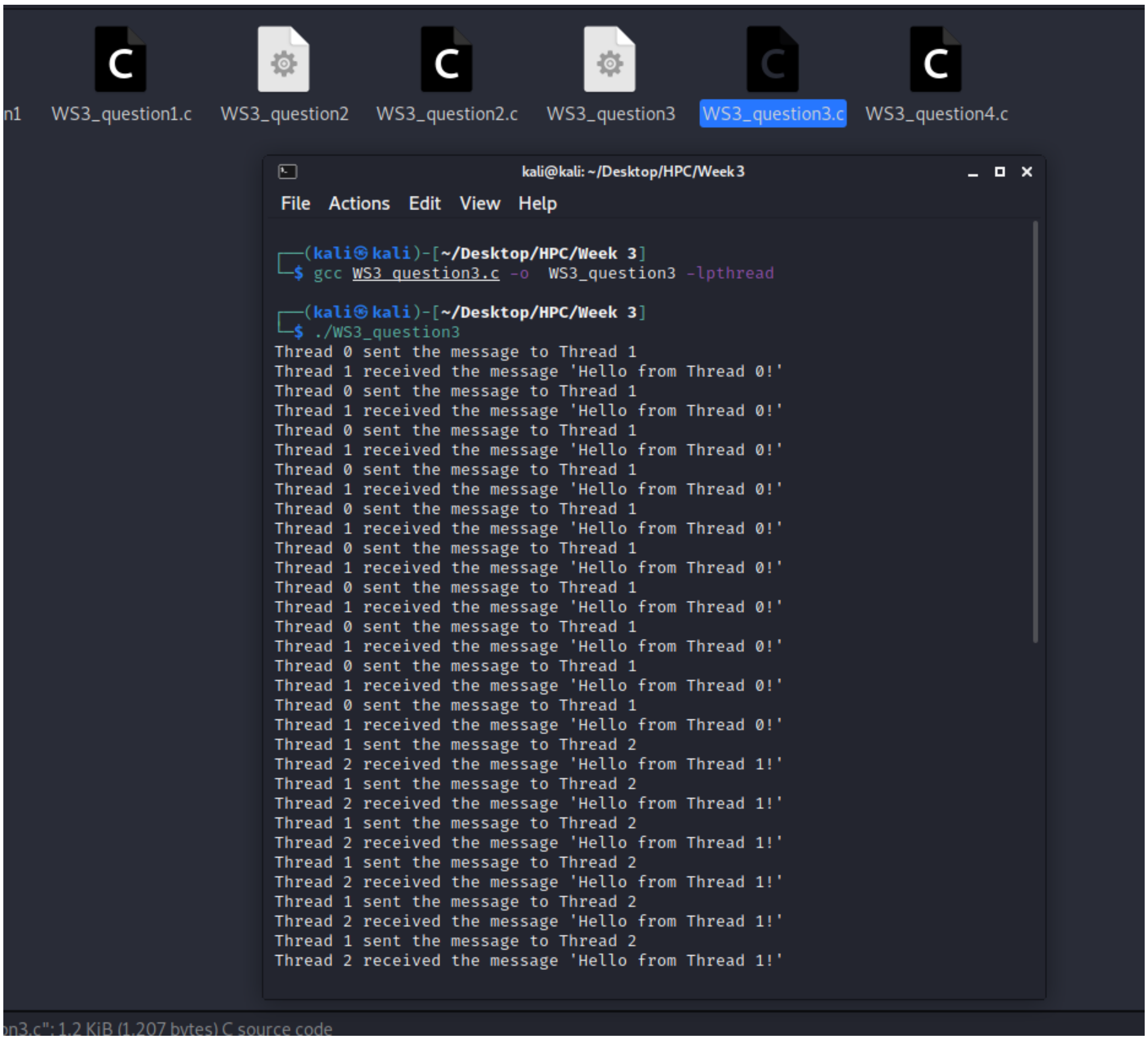
int flag = 0;

pthread_mutex_t mutex;

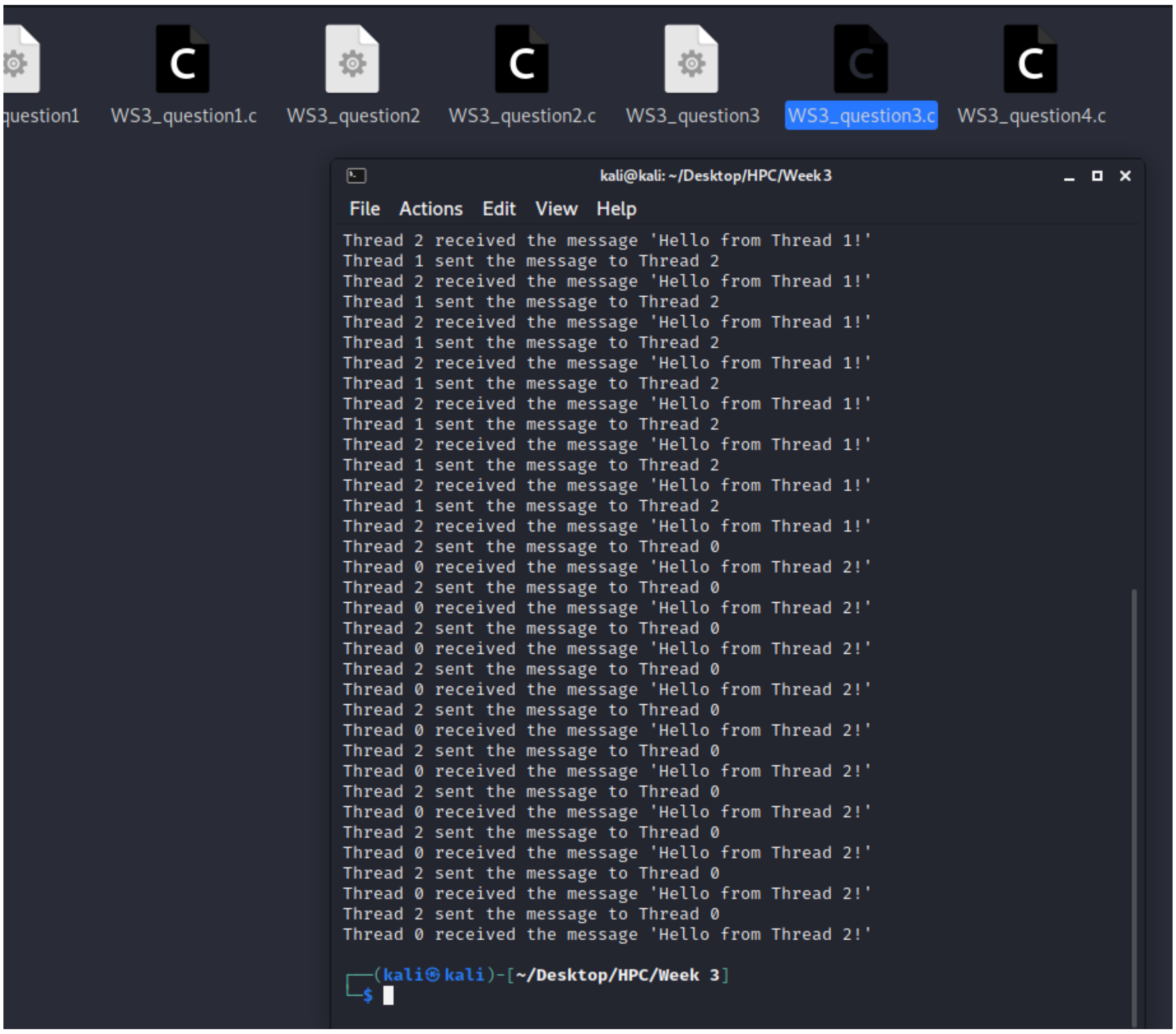
```
void *messenger(void *p)
{
   long tid = (long)p;
   char tmpbuf[100];
      for(int i=0; i<10; i++)
      {
         pthread_mutex_lock(&mutex);

         /* Sending a message */

         long int dest = (tid + 1) % 3;
         sprintf(tmpbuf,"Hello from Thread %ld!", tid);
         char *msg = strdup(tmpbuf);
         messages[dest] = msg;
         printf("Thread %ld sent the message to Thread %ld\n",tid, dest);

         /* Receiving a message */

         printf("Thread %ld received the message '%s'\n",dest, messages[dest]);
         free(messages[dest]);
         messages[dest] = NULL;
         pthread_mutex_unlock(&mutex);
      }
   return NULL;
}



void main()
{
   pthread_t thrID1, thrID2, thrID3;
```

```
    pthread_create(&thrID1, NULL, messenger, (void *)0);
    pthread_create(&thrID2, NULL, messenger, (void *)1);
    pthread_create(&thrID3, NULL, messenger, (void *)2);
    pthread_join(thrID1, NULL);
    pthread_join(thrID2, NULL);
    pthread_join(thrID3, NULL);
}
```

OUTPUT: -


n1
WS3_question1.c
WS3_question2
WS3_question2.c
WS3_question3
WS3_question3.c
WS3_question4.c
kali@kali: ~/Desktop/HPC/Week3
File Actions Edit View Help
on3.c": 1.2 KiB (1,207 bytes) C source code

```
┌──(kali㉿kali)-[~/Desktop/HPC/Week 3]
└─$ gcc WS3_question3.c -o  WS3_question3 -lpthread

┌──(kali㉿kali)-[~/Desktop/HPC/Week 3]
└─$ ./WS3_question3
Thread 0 sent the message to Thread 1
Thread 1 received the message 'Hello from Thread 0!'
Thread 0 sent the message to Thread 1
Thread 1 received the message 'Hello from Thread 0!'
Thread 0 sent the message to Thread 1
Thread 1 received the message 'Hello from Thread 0!'
Thread 0 sent the message to Thread 1
Thread 1 received the message 'Hello from Thread 0!'
Thread 0 sent the message to Thread 1
Thread 1 received the message 'Hello from Thread 0!'
Thread 0 sent the message to Thread 1
Thread 1 received the message 'Hello from Thread 0!'
Thread 0 sent the message to Thread 1
Thread 1 received the message 'Hello from Thread 0!'
Thread 0 sent the message to Thread 1
Thread 1 received the message 'Hello from Thread 0!'
Thread 0 sent the message to Thread 1
Thread 1 received the message 'Hello from Thread 0!'
Thread 0 sent the message to Thread 1
Thread 1 received the message 'Hello from Thread 0!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
```



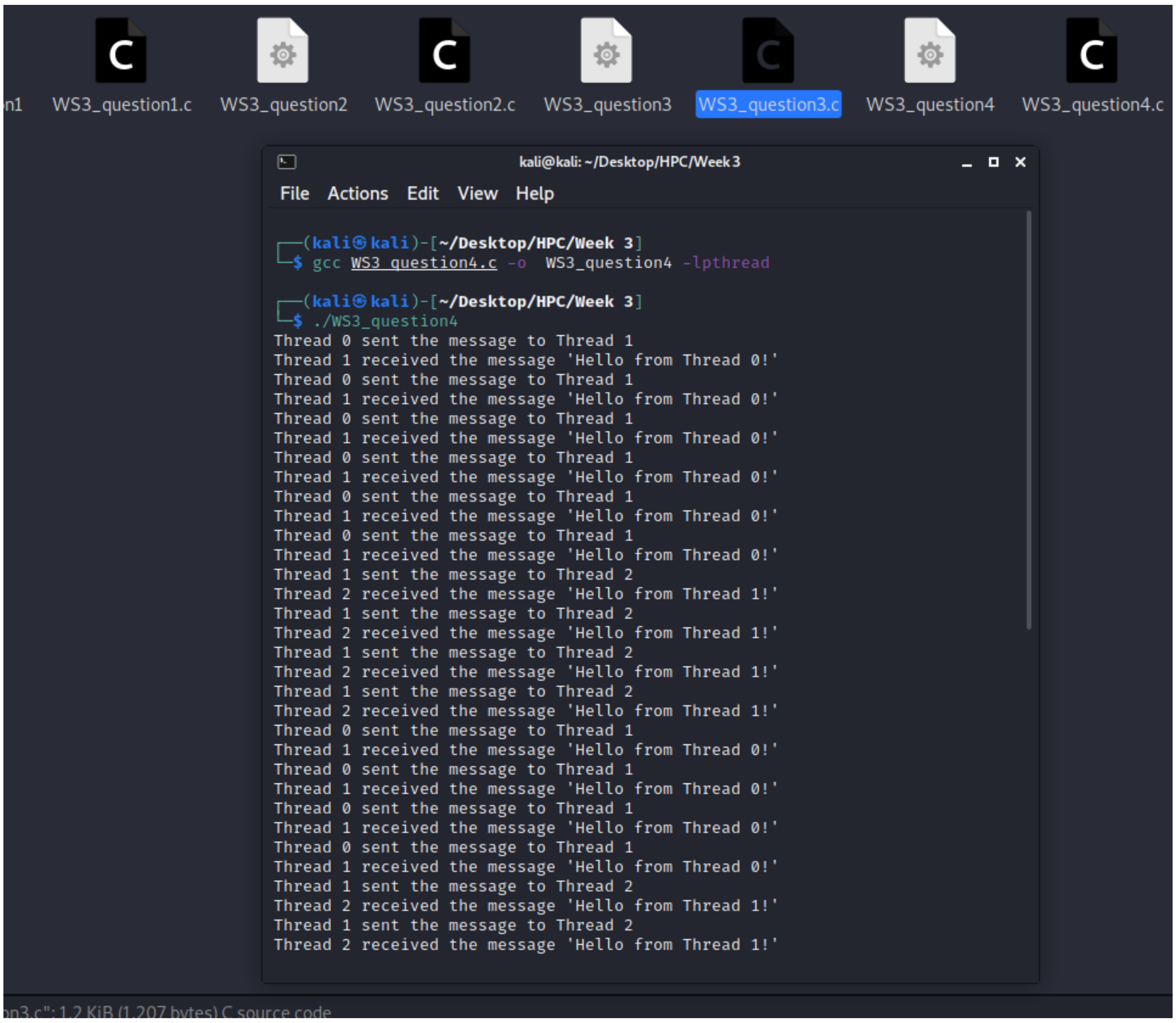
```
Thread 2 received the message 'Hello from Thread 1!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 2 sent the message to Thread 0
Thread 0 received the message 'Hello from Thread 2!'
Thread 2 sent the message to Thread 0
Thread 0 received the message 'Hello from Thread 2!'
Thread 2 sent the message to Thread 0
Thread 0 received the message 'Hello from Thread 2!'
Thread 2 sent the message to Thread 0
Thread 0 received the message 'Hello from Thread 2!'
Thread 2 sent the message to Thread 0
Thread 0 received the message 'Hello from Thread 2!'
Thread 2 sent the message to Thread 0
Thread 0 received the message 'Hello from Thread 2!'
Thread 2 sent the message to Thread 0
Thread 0 received the message 'Hello from Thread 2!'
Thread 2 sent the message to Thread 0
Thread 0 received the message 'Hello from Thread 2!'
Thread 2 sent the message to Thread 0
Thread 0 received the message 'Hello from Thread 2!'
Thread 2 sent the message to Thread 0
Thread 0 received the message 'Hello from Thread 2!'

┌──(kali㉿kali)-[~/Desktop/HPC/Week 3]
└─$ 
```

4. Use semaphores to correct the program in (1).

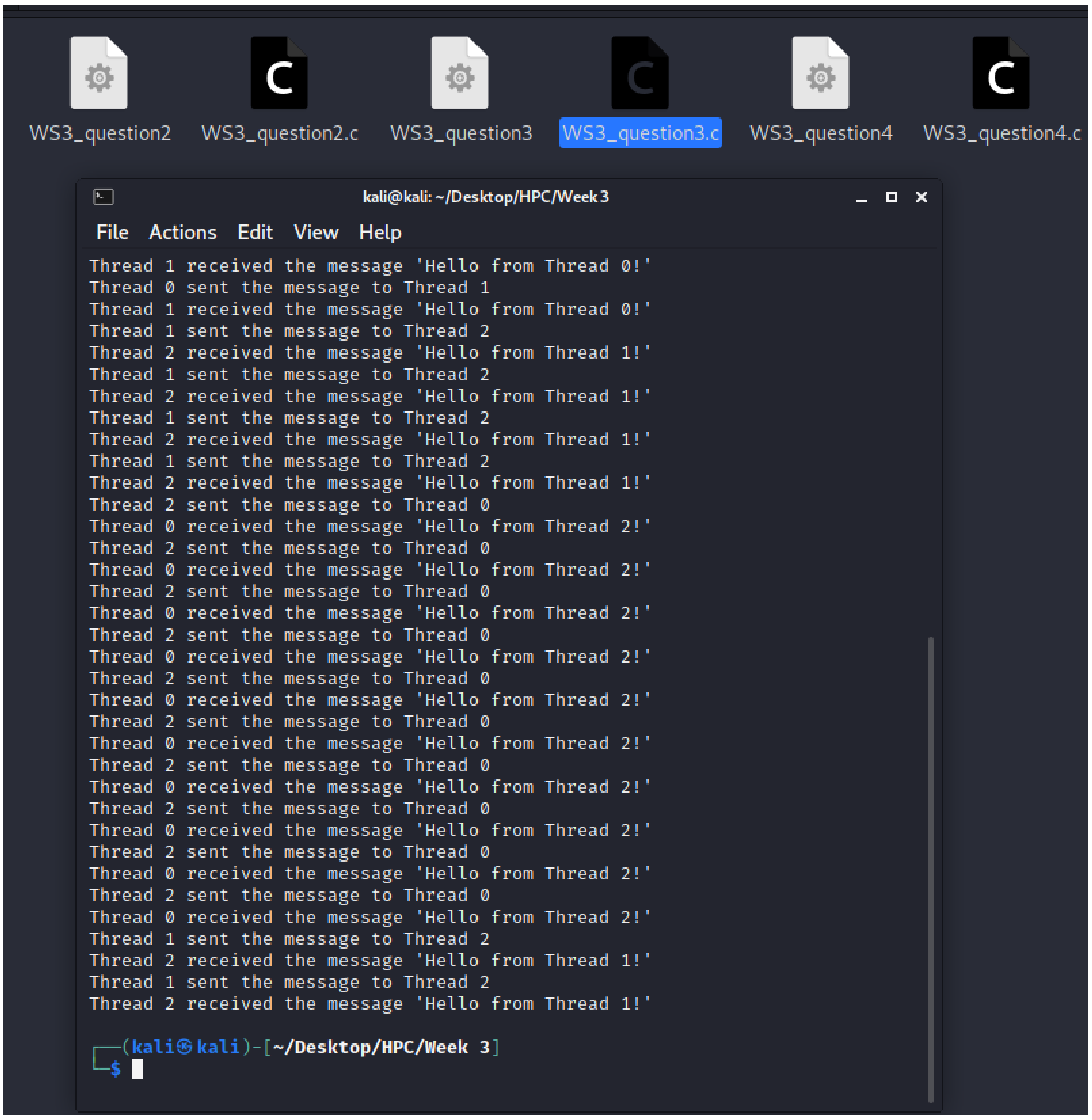
```
=> #include <stdio.h>
   #include <stdlib.h>
   #include <string.h>
   #include <pthread.h>
   #include <unistd.h>
  #include<semaphore.h>

char *messages[3] = {NULL, NULL, NULL};
int flag = 0;
sem_t lock;

void *messenger(void *p)
```

```
{
    long tid = (long)p;
    char tmpbuf[100];
        for(int i=0; i<10; i++)
        {
            sem_wait(&lock);
            /* Sending a message */
            long int dest = (tid + 1) % 3;
            sprintf(tmpbuf,"Hello from Thread %ld!", tid);
            char *msg = strdup(tmpbuf);
            messages[dest] = msg;
            printf("Thread %ld sent the message to Thread %ld\n",tid, dest);



            /* Receiving a message */


            printf("Thread %ld received the message '%s'\n",dest, messages[dest]);
            free(messages[dest]);
            messages[dest] = NULL;
            sem_post(&lock);
        }
    return NULL;
}
void main()
{
    pthread_t thrID1, thrID2, thrID3;
    sem_init(&lock, 1, 1);
    pthread_create(&thrID1, NULL, messenger, (void *)0);
    pthread_create(&thrID2, NULL, messenger, (void *)1);
    pthread_create(&thrID3, NULL, messenger, (void *)2);
    pthread_join(thrID1, NULL);
    pthread_join(thrID2, NULL);
    pthread_join(thrID3, NULL);
    sem_destroy(&lock);
}
```

OUTPUT: -



```
┌──(kali㉿kali)-[~/Desktop/HPC/Week 3]
└─$ gcc WS3_question4.c -o  WS3_question4 -lpthread

┌──(kali㉿kali)-[~/Desktop/HPC/Week 3]
└─$ ./WS3_question4
Thread 0 sent the message to Thread 1
Thread 1 received the message 'Hello from Thread 0!'
Thread 0 sent the message to Thread 1
Thread 1 received the message 'Hello from Thread 0!'
Thread 0 sent the message to Thread 1
Thread 1 received the message 'Hello from Thread 0!'
Thread 0 sent the message to Thread 1
Thread 1 received the message 'Hello from Thread 0!'
Thread 0 sent the message to Thread 1
Thread 1 received the message 'Hello from Thread 0!'
Thread 0 sent the message to Thread 1
Thread 1 received the message 'Hello from Thread 0!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 0 sent the message to Thread 1
Thread 1 received the message 'Hello from Thread 0!'
Thread 0 sent the message to Thread 1
Thread 1 received the message 'Hello from Thread 0!'
Thread 0 sent the message to Thread 1
Thread 1 received the message 'Hello from Thread 0!'
Thread 0 sent the message to Thread 1
Thread 1 received the message 'Hello from Thread 0!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
```



```
Thread 1 received the message 'Hello from Thread 0!'
Thread 0 sent the message to Thread 1
Thread 1 received the message 'Hello from Thread 0!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 2 sent the message to Thread 0
Thread 0 received the message 'Hello from Thread 2!'
Thread 2 sent the message to Thread 0
Thread 0 received the message 'Hello from Thread 2!'
Thread 2 sent the message to Thread 0
Thread 0 received the message 'Hello from Thread 2!'
Thread 2 sent the message to Thread 0
Thread 0 received the message 'Hello from Thread 2!'
Thread 2 sent the message to Thread 0
Thread 0 received the message 'Hello from Thread 2!'
Thread 2 sent the message to Thread 0
Thread 0 received the message 'Hello from Thread 2!'
Thread 2 sent the message to Thread 0
Thread 0 received the message 'Hello from Thread 2!'
Thread 2 sent the message to Thread 0
Thread 0 received the message 'Hello from Thread 2!'
Thread 2 sent the message to Thread 0
Thread 0 received the message 'Hello from Thread 2!'
Thread 2 sent the message to Thread 0
Thread 0 received the message 'Hello from Thread 2!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'
Thread 1 sent the message to Thread 2
Thread 2 received the message 'Hello from Thread 1!'

┌──(kali㉿kali)-[~/Desktop/HPC/Week 3]
└─$ 
```