```
import threading
import time

def writeFile(msg, output_fname):
  fh = open(output_fname, "w")
  fh.write(msg + "\n")
  time.sleep(5)
  fh.close()

class SyncWrite(threading.Thread):
  def __init__(self, msg, output_fname):
    threading.Thread.__init__(self)
    self.msg = msg
    self.output_fname = output_fname

  def run(self):
    # Acquire the Lock
    trdLock.acquire()

    print ("Writing the contents to the file -> " + self.output_fname)
    writeFile(self.msg, self.output_fname)
    print ("Finished writing the contents to the file -> " +
self.output_fname)

    # Release the Lock
    trdLock.release()

# Creating the Thread Lock object - which will make the process
synchronous
trdLock = threading.Lock()

# Empty threads to store all the threads
threads = []

# Retrieve the messages
msg1 = input("Enter Message 1 -> ")
msg2 = input("Enter Message 2 -> ")

# Create the Threads
trd1 = SyncWrite(msg1, "thread1.txt")
trd2 = SyncWrite(msg2, "thread2.txt")

# Append the thread to the list
threads.append(trd1)
threads.append(trd2)

# Starting the Threads
trd1.start()
trd2.start()
```

```
# Waiting the program to complete the execution of threads
for trd in threads:
  trd.join()

print ("Waited until both the threads writing the contents to the file")
```