

```
Thread-2: Thu Mar 21 09:10:08 2013  
Thread-2: Thu Mar 21 09:10:10 2013  
Thread-2: Thu Mar 21 09:10:12 2013  
Exiting Thread-2
```

Synchronizing Threads

The `threading` module provided with Python includes a simple-to-implement locking mechanism that allows you to synchronize threads. A new lock is created by calling the `Lock()` method, which returns the new lock.

The `acquire(blocking)` method of the new lock object is used to force threads to run synchronously. The optional `blocking` parameter enables you to control whether the thread waits to acquire the lock.

If `blocking` is set to 0, the thread returns immediately with a 0 value if the lock cannot be acquired and with a 1 if the lock was acquired. If `blocking` is set to 1, the thread blocks and wait for the lock to be released.

The `release()` method of the new lock object is used to release the lock when it is no longer required.

Example

```
#!/usr/bin/python  
  
import threading  
import time  
  
class myThread (threading.Thread):  
    def __init__(self, threadID, name, counter):  
        threading.Thread.__init__(self)  
        self.threadID = threadID  
        self.name = name  
        self.counter = counter
```