CSC543 students, here is an overview of using CONTROL-\
(control backslash)
To get stack traces of your threads if **make test** hangs.

$ **grep sleep *java**
TupleExchange.java: Thread.sleep(10000); // This is the source of my demo hang.

[:-) ~/.../solutions/CSC543s24DataflowAssn1] **make test**
STUFF DELETED…
bash -c "CLASSPATH=..:./jcip-annotations.jar time /usr/bin/java
CSC543s24DataflowAssn1.CSC543s24DataflowAssn1 343458543 test1.csv 0"
The single-threaded test passes…
STUFF DELETED…
diff --ignore-trailing-space --strip-trailing-cr CSC543s24single.csv CSC543s24single.ref >
CSC543s24single.dif

bash -c "CLASSPATH=..:./jcip-annotations.jar time /usr/bin/java
CSC543s24DataflowAssn1.CSC543s24DataflowAssn1 343458543 test2.csv 1"
The multi-threaded test hangs, so I hit control-\:
^
2024-02-05 17:41:25
Full thread dump OpenJDK 64-Bit Server VM (25.322-b06 mixed mode):
:
Ignore everything except the stack traces except for thread main and Thread-N, where N is one
of your project threads. There are 5, N=0 thru 4, in this assignment.

"**Thread-4**" #25 prio=5 os_prio=0 tid=0x00007f589c1b2800 nid=0x2fb7 waiting on condition
[0x00007f581480b000]
   java.lang.Thread.State: TIMED_WAITING (sleeping)
      at **java.lang.Thread.sleep**(Native Method)
      at **CSC543s24DataflowAssn1.TupleExchange.consumeTenTuple(TupleExchange.java:117)**
      at
CSC543s24DataflowAssn1.StatisticalAnalysisCSVSaver.run1iteration(StatisticalAnalysisCSVSaver.java:63)
      at
CSC543s24DataflowAssn1.StatisticalAnalysisCSVSaver.run(StatisticalAnalysisCSVSaver.java:58)
      at java.lang.Thread.run(Thread.java:750)
****
"**Thread-3**" #24 prio=5 os_prio=0 tid=0x00007f589c1b0800 nid=0x2fb6 waiting on condition
[0x00007f581490c000]
   java.lang.Thread.State: WAITING (parking)
      at sun.misc.Unsafe.park(Native Method)

- parking to wait for <0x0000000671e73f40>

java.util.concurrent.locks.ReentrantLock$NonfairSync) // It is waiting to get the lock.at
java.util.concurrent.locks.LockSupport.park(LockSupport.java:175)
at
java.util.concurrent.locks.AbstractQueuedSynchronizer.parkAndCheckInterrupt(AbstractQueue
dSynchronizer.java:836)
 at
java.util.concurrent.locks.AbstractQueuedSynchronizer.acquireQueued(AbstractQueuedSynchr
onizer.java:870)
at
java.util.concurrent.locks.AbstractQueuedSynchronizer.acquire(AbstractQueuedSynchronizer.ja
va:1199)
at java.util.concurrent.locks.ReentrantLock$NonfairSync.lock(ReentrantLock.java:209)
at java.util.concurrent.locks.**ReentrantLock.lock**(ReentrantLock.java:285)
at CSC543s24DataflowAssn1.**TupleExchange.produceTenTuple(TupleExchange.java:95)**
at
**CSC543s24DataflowAssn1.StatisticalAnalysisGenerator.run1iteration(StatisticalAnalysisGener
ator.java:53**)
at
CSC543s24DataflowAssn1.StatisticalAnalysisGenerator.run(StatisticalAnalysisGenerator.java:39
)
at java.lang.Thread.run(Thread.java:750)
****

 "**Thread-2**" #23 prio=5 os_prio=0 tid=0x00007f589c1af000 nid=0x2fb5 in **Object.wait**()
[0x00007f5814a0d000]
java.lang.Thread.State: WAITING (on object monitor)
at java.lang.Object.wait(Native Method)
- waiting on <0x0000000671e72bf8> (a CSC543s24DataflowAssn1.TupleExchange)
at java.lang.Object.wait(Object.java:502)
at CSC543s24DataflowAssn1.**TupleExchange.produceFiveTuple(TupleExchange.java:67)**
- locked <0x0000000671e72bf8> (a CSC543s24DataflowAssn1.TupleExchange)
at
CSC543s24DataflowAssn1.**DistributionGenerator.run1iteration(DistributionGenerator.java:45)**
at CSC543s24DataflowAssn1.DistributionGenerator.run(DistributionGenerator.java:39)
at java.lang.Thread.run(Thread.java:750)
****

 "**Thread-1**" #22 prio=5 os_prio=0 tid=0x00007f589c1ad000 nid=0x2fb4 in Object.wait()
[0x00007f5814b0e000]
   java.lang.Thread.State: WAITING (on object monitor)
at java.lang.Object.wait(Native Method)
- waiting on <0x0000000671e72bf8> (a CSC543s24DataflowAssn1.TupleExchange)
at **java.lang.Object.wait**(Object.java:502)
at **CSC543s24DataflowAssn1.TupleExchange.produceFiveTuple(TupleExchange.java:67)**
- locked <0x0000000671e72bf8> (a CSC543s24DataflowAssn1.TupleExchange)

at CSC543s24DataflowAssn1.DistributionGenerator.run1iteration(DistributionGenerator.java:45)
at CSC543s24DataflowAssn1.DistributionGenerator.run(DistributionGenerator.java:39)
at java.lang.Thread.run(Thread.java:750)
****

"**Thread-0**" #21 prio=5 os_prio=0 tid=0x00007f589c1ab000 nid=0x2fb3 in Object.wait() [0x00007f5814c0f000]
java.lang.Thread.State: WAITING (on object monitor)
at **java.lang.Object.wait**(Native Method)
- waiting on <0x0000000671e72bf8> (a CSC543s24DataflowAssn1.TupleExchange)
at java.lang.Object.wait(Object.java:502)
at CSC543s24DataflowAssn1.TupleExchange.produceFiveTuple(TupleExchange.java:67)
- locked <0x0000000671e72bf8> (a CSC543s24DataflowAssn1.TupleExchange)
at CSC543s24DataflowAssn1.DistributionGenerator.run1iteration(DistributionGenerator.java:45)
at CSC543s24DataflowAssn1.DistributionGenerator.run(DistributionGenerator.java:39)
at java.lang.Thread.run(Thread.java:750)
****

 IGNORE OTHERS UNTIL MAIN THREAD:
****

"main" #1 prio=5 os_prio=0 tid=0x00007f589c04b800 nid=0x2f86 in Object.wait() [0x00007f58a4a7a000]
java.lang.Thread.State: WAITING (on object monitor)
at java.lang.Object.wait(Native Method)
- waiting on <0x0000000671e82148> (a java.lang.Thread)
at **java.lang.Thread.join**(Thread.java:1257)
- locked <0x0000000671e82148> (a java.lang.Thread)
at **java.lang.Thread.join**(Thread.java:1331)
at **CSC543s24DataflowAssn1.CSC543s24DataflowAssn1.main(CSC543s24DataflowAssn1.java:83)**
"VM Thread" os_prio=0 tid=0x00007f589c101000 nid=0x2f9e runnable
****

 Ignore the GC garbage collector and compile threads and everything else.
Above are the stack traces including line numbers for your assignment threads.