Homework 1

CSE 461 – Yousef Jarrar

Dr. Tong Yu

1. ( 10 points ) \*\*Extra Credit for 461 Students\*\*   
We have discussed in the class the implementation of the readers-writers problem in Java. However, the read and write tasks of the **reader** thread and the **writer** thread are not given. Implement these tasks in Java as reading and writing of a file named *counter.txt*, which contains an integer counter.   
A **reader** thread

reads the counter from the file, and

prints out its thread name and the value of the counter.

A **writer** thread

increments the value of the counter in the file,

prints out its thread name and the new value of the counter.

Each thread repeats its task indefinitely in a random amount of time between 0 and 3000 ms. Your **main** program should create 20 **reader** threads and 3 **writer** threads.

Besides the source code, turn in scripts showing that you compile and run the program successfully. Turn in also some sample outputs.

/\*

\* Class created to hold lock and condition

\* \*/

import java.io.File;

import java.io.FileInputStream;

import java.io.FileWriter;

import java.io.IOException;

import java.util.Scanner;

import java.util.concurrent.locks.Condition;

import java.util.concurrent.locks.Lock;

import java.util.concurrent.locks.ReentrantLock;

public class ReadWrites {

private final String file = "counter.txt";

private final Lock \_mutex = new ReentrantLock();// create mutex instance

private final Condition readerQueue = \_mutex.newCondition();// Returns a new condition for

// reader that is bound to this Lock instance.

private final Condition writerQueue = \_mutex.newCondition();// Returns a new condition for writer that

// is bound to this Lock instance.

private int readers\_count = 0;// to store current readers count

private int writers\_count = 0;// to store current writers count

public ReadWrites() {

try {

FileWriter filewriter = new FileWriter(new File(file));

filewriter.write(new Integer(0).toString());

filewriter.close();

} catch (IOException e) {

e.printStackTrace();

}

}

void readerrun() throws InterruptedException {

\_mutex.lock(); // acquire lock

while (!(writers\_count == 0)) {

readerQueue.await();

}

readers\_count++;

\_mutex.unlock();

read(file);//read file

\_mutex.lock();

if (--readers\_count == 0) {

writerQueue.signal();//signal writers

}

\_mutex.unlock();//remove lock

}

void writerrun() throws InterruptedException {

\_mutex.lock();

while (!((readers\_count == 0) && (writers\_count == 0))) {

writerQueue.await();// when reader and writer is zero wait in

// writerQueue

}

writers\_count++; // increment writer

\_mutex.unlock();// remove lock

write(file);// write to file

\_mutex.lock(); // acquire lock

writers\_count--; // only one writer at a time

writerQueue.signal(); // signal writers

readerQueue.signalAll(); // signal all readers

\_mutex.unlock();// remove lock

}

void read(String path) {

try {

Scanner reader = new Scanner(new FileInputStream(path));

int x = reader.nextInt();

System.out.printf(Thread.currentThread().getName() + " is reading ...");

System.out.printf(" Counter value : %d\n", x);

reader.close();

} catch (IOException ex) {

ex.printStackTrace();

}

}

void write(String path) {

int countw;

try {

Scanner reader = new Scanner(new FileInputStream(path));

countw = (int) reader.nextInt();

countw++;

FileWriter f = new FileWriter(new File(path));

f.write(new Integer(countw).toString());

f.close();

System.out.printf(Thread.currentThread().getName() + " Writing... ");

System.out.printf(" Counter value : %d\n", countw);

reader.close();

} catch (IOException ex) {

ex.printStackTrace();

}

}

}

**OUTPUT:**

**Program Start**

**Reader Thread-21: Started**

**Reader Thread-10: Started**

**Reader Thread-4: Started**

**Reader Thread-14: Started**

**Reader Thread-20: Started**

**Reader Thread-16: Started**

**Reader Thread-5: Started**

**Reader Thread-8: Started**

**Reader Thread-9: Started**

**Reader Thread-7: Started**

**Reader Thread-19: Started**

**Writer Thread-1: Started**

**Reader Thread-15: Started**

**Reader Thread-11: Started**

**Writer Thread-0: Started**

**Reader Thread-13: Started**

**Reader Thread-17: Started**

**Reader Thread-6: Started**

**Writer Thread-2: Started**

**Reader Thread-22: Started**

**Reader Thread-12: Started**

**Reader Thread-18: Started**

**Reader Thread-3: Started**

**Thread-7 is reading ...Thread-15 is reading ... Counter value : 0**

**Thread-5 is reading ... Counter value : 0**

**Thread-13 is reading ... Counter value : 0**

**Thread-10 is reading ... Counter value : 0**

**Thread-20 is reading ... Counter value : 0**

**Thread-8 is reading ... Counter value : 0**

**Thread-18 is reading ... Counter value : 0**

**Thread-17 is reading ... Counter value : 0**

**Thread-22 is reading ... Counter value : 0**

**Thread-6 is reading ... Counter value : 0**

**Thread-19 is reading ... Counter value : 0**

**Thread-3 is reading ... Counter value : 0**

**Thread-14 is reading ... Counter value : 0**

**Thread-9 is reading ... Counter value : 0**

**Thread-16 is reading ... Counter value : 0**

**Thread-21 is reading ... Counter value : 0**

**Thread-4 is reading ... Counter value : 0**

**Thread-12 is reading ... Counter value : 0**

**Thread-11 is reading ... Counter value : 0**

**Counter value : 0**

**Thread-1 Writing... Counter value : 1**

**Thread-0 Writing... Counter value : 2**

**Thread-2 Writing... Counter value : 3**

**Thread-4 is reading ... Counter value : 3**

**Thread-0 Writing... Counter value : 4**

**Thread-5 is reading ... Counter value : 4**

**Thread-19 is reading ... Counter value : 4**

**Thread-8 is reading ... Counter value : 4**

**Thread-9 is reading ... Counter value : 4**

**Thread-14 is reading ... Counter value : 4**

**Thread-9 is reading ... Counter value : 4**

**Thread-13 is reading ... Counter value : 4**

**Thread-2 Writing... Counter value : 5**

**Thread-19 is reading ... Counter value : 5**

**Thread-18 is reading ... Counter value : 5**

**Thread-17 is reading ... Counter value : 5**

**Thread-3 is reading ... Counter value : 5**

**Thread-20 is reading ... Counter value : 5**

**Thread-11 is reading ... Counter value : 5**

**Thread-15 is reading ... Counter value : 5**

**Thread-6 is reading ... Counter value : 5**

**Thread-1 Writing... Counter value : 6**

**Thread-13 is reading ... Counter value : 6**

**Thread-19 is reading ... Counter value : 6**

**Thread-17 is reading ... Counter value : 6**

**Thread-8 is reading ... Counter value : 6**

**Thread-16 is reading ... Counter value : 6**

**Thread-10 is reading ... Counter value : 6**

**Thread-14 is reading ... Counter value : 6**

**Thread-3 is reading ... Counter value : 6**

**Thread-16 is reading ... Counter value : 6**

**Thread-22 is reading ... Counter value : 6**

**Thread-3 is reading ... Counter value : 6**

**Thread-2 Writing... Counter value : 7**

**Thread-12 is reading ... Counter value : 7**

**Thread-21 is reading ... Counter value : 7**

**Thread-7 is reading ... Counter value : 7**

**Thread-17 is reading ... Counter value : 7**

**Thread-4 is reading ... Counter value : 7**

**Thread-9 is reading ... Counter value : 7**

**Thread-7 is reading ... Counter value : 7**

**Thread-5 is reading ... Counter value : 7**

**Thread-11 is reading ... Counter value : 7**

**Thread-2 Writing... Counter value : 8**

**Thread-1 Writing... Counter value : 9**

**Thread-7 is reading ... Counter value : 9**

**Thread-16 is reading ... Counter value : 9**

**Thread-0 Writing... Counter value : 10**

**Thread-11 is reading ... Counter value : 10**

**Thread-0 Writing... Counter value : 11**

**Thread-20 is reading ... Counter value : 11**

**Thread-0 Writing... Counter value : 12**

**Thread-21 is reading ... Counter value : 12**

**Thread-6 is reading ... Counter value : 12**

**Thread-9 is reading ... Counter value : 12**

**Thread-18 is reading ... Counter value : 12**

**Thread-13 is reading ... Counter value : 12**

**Thread-2 Writing... Counter value : 13**

**Thread-3 is reading ... Counter value : 13**

**Thread-8 is reading ... Counter value : 13**

**Thread-8 is reading ... Counter value : 13**

**Thread-14 is reading ... Counter value : 13**

**Thread-22 is reading ... Counter value : 13**

**Thread-15 is reading ... Counter value : 13**

**Thread-20 is reading ... Counter value : 13**

**Thread-19 is reading ... Counter value : 13**

**Thread-8 is reading ... Counter value : 13**

**Thread-5 is reading ... Counter value : 13**

**Thread-14 is reading ... Counter value : 13**

**Thread-13 is reading ... Counter value : 13**

**Thread-10 is reading ... Counter value : 13**

**Thread-7 is reading ... Counter value : 13**

**Thread-11 is reading ... Counter value : 13**

**Thread-21 is reading ... Counter value : 13**

**Thread-16 is reading ... Counter value : 13**

**Thread-18 is reading ... Counter value : 13**

6.

7. Consider a chain of processes P1, P2, ..., Pn implementing a multitiered client-server architecture. Process Pi is client of process Pi+1, and Pi will return a reply to Pi-1 only after receiving a reply from Pi+1. What are the main problems with this organization when taking a look at the request-reply performance at process P1?