**Lab Book**

**Exercise 2.1**

**Diagram:**

Connect2

Connect1

Multiplier

Consumer

Producer

**Code:**

**def** processList =

[

**new** Producer ( outChannel: connect1.out() ),

//insert here an instance of multiplier with a multiplication factor of 4

**new** Multiplier(inChannel: connect1.in(), factor: 4, outChannel: connect2.out()),

**new** Consumer ( inChannel: connect2.in() )

]

**while** (i > 0)

{

// write i \* factor to outChannel

outChannel.write(i\*factor);

// read in the next value of i

i = inChannel.read();

}

**while** ( i > 0 )

{

//insert a modified println statement

println i;

i = inChannel.read()

}

**Output:**

3

next: 12

5

next: 20

6

next: 24

**Exercise 2.2**

**Diagram:**

Connect2

Connect1

CreateSetsOfEight

ListToStream

GenerateSetsOfThree

**Code:**

**GenerateSetsOfThree**

//write the terminating List as per exercise definition

outChannel.write([-1,-1,-1])

**ListToStream**

// hint: output list elements as single integers

**for**(j **in** 0..<inList.size)

{

outChannel.write(inList[j])

}

inList = inChannel.read()

**CreateSetsOfEight**

**while** (v != -1)

{

**for** ( i **in** 0 .. 7 )

{

// put v into outList and read next input

outList.add(v)

v = inChannel.read()

}

*println* " Eight Object is ${outList}"

outList.clear()

}

**Output:**

Eight Object is [1, 2, 3, 4, 5, 6, 7, 8]

Eight Object is [9, 10, 11, 12, 13, 14, 15, 16]

Eight Object is [17, 18, 19, 20, 21, 22, 23, 24]

Finished

**Questions:**

What change is required to output objects containing six integers?

Change the

**for** ( i **in** 0 .. 7 )

in CreateSetsOfEight to

**for** ( i **in** 0 .. 5 )

How could you paramaterise this in the system to output objects that contain any number of integers?

Instead of changing it to a static value, create a variable setSize and change the loop to:

**for** ( i **in** 0 .. (setSize – 1) )

What happens if the number of integers required in the output stream is not a factor of the total number of integers in the input stream?

The numbers that go beyond the maximum factor are excluded from the output stream.

**Exercise 3.1**

**Minus:**

**Diagram:**

out

in

a

c

Minus

GPrefix

b

GPCopy

**Code:**

**Minus.groovy**

outChannel.write(read0.value - read1.value)

**Differentiate.groovy**

**def** differentiateList = [ **new** GPrefix(prefixValue: 0,

inChannel: b.**in**(),

outChannel: c.out() ),

**new** GPCopy ( inChannel: inChannel,

outChannel0: a.out(),

outChannel1: b.out() ),

// insert a constructor for Minus

**new** Minus ( inChannel0: a.**in**(),

inChannel1: c.**in**(),

outChannel: outChannel)

]

**Output:**

Differentiated Numbers

0

1

2

3

4

5

6

7

**Negator:**

**Diagram:**

out

a

in

GPCopy

d

c

b

GPlus

GPrefix

Negator

**Code:**

**Negator.groovy**

outChannel.write(-inChannel.read())

**DifferentiateNeg.groovy**

**def** differentiateList = [ **new** GPrefix ( prefixValue: 0,

inChannel: b.**in**(),

outChannel: c.out() ),

**new** GPCopy ( inChannel: inChannel, outChannel0: a.out(),

outChannel1: b.out() ),

//insert a constructor for Negator

**new** Negator ( inChannel: c.**in**(),

outChannel: d.out()),

**new** GPlus ( inChannel0: a.**in**(),

inChannel1: d.**in**(),

outChannel: outChannel )

]

**Output:**

Differentiated Numbers

0

1

2

3

4

5

6

7

**Questions:**

I find that the minus method is more pleasing, as it is more sensible to have a method for subtracting than it is to negate the value and then add it.

Exercise 3.2

**Diagram:**

GSPairsA

a

out

in

c

GPlus

b

GSCopy

GTail

GSPairsB

GTail

in

b

c

out

a

GPlus

GSCopy

**Code:**

**GSCopy**

**while** (**true**)

{

**def** i = inChannel.read()

// output the input value in sequence to each output channel

outChannel0.write(i)

outChannel1.write(i)

}

**GSquares**

**With GSPairsA:**

**def** testList = [ **new** GNumbers ( outChannel: N2I.out() ),

**new** GIntegrate ( inChannel: N2I.**in**(),

outChannel: I2P.out() ),

**new** GSPairsA(inChannel: I2P.**in**(),

outChannel: outChannel)

]

**With GSPairsB:**

**def** testList = [ **new** GNumbers ( outChannel: N2I.out() ),

**new** GIntegrate ( inChannel: N2I.**in**(),

outChannel: I2P.out() ),

**new** GSPairsB(inChannel: I2P.**in**(),

outChannel: outChannel)

]

Output:

With GSPairsA:

Squares

With GSPairsB:

Squares

1

4

9

16

25

36

49

**Questions:**

When GSPairsA is used the channel a is sent first from GSCopy, and reaches GPlus, which is in turn waiting for the other in channel, which will never arrive as GSCopy is still sending through a, as such the process never completes. In GSPairsB the b channel is sent first, reaches GTail which then sends out c which reaches GPlus, which then waits for the a channel, which is available.

**Exercise 3.3:**

GParprint outputs the data in order while GPrint would not, as such the GParPrint method will make it easier to present the data in a table.

**Exercise 4.1:**

**Questions:**

If line 25 is removed the previous value will continue to be printed along with the new value. This occurs because the previous value still exists in the channel and must be read in order to be discarded.

When multiple reset values are entered the program deadlocks. This is because the number of values being entered from the reset channel outnumber those coming from GSuccessor. This works when there are two values in the system because there will always be a free process, but when more are added each process will be used and the deadlock occurs.

**Exercise 4.2:**

Diagram:

in

a

out

GSCopy

GPrefix

Code:

resetChannel

b

c

ResetSuccessor

**ResetNumbers:**

**def** testList = [ **new** GPrefix ( prefixValue: initialValue,

outChannel: a.out(),

inChannel: c.**in**() ),

**new** GPCopy ( inChannel: a.**in**(),

outChannel0: outChannel,

outChannel1: b.out() ),

// requires a constructor for ResetSuccessor

**new** ResetSuccessor( resetChannel: resetChannel,

inChannel: b.**in**(),

outChannel: c.out())

]

**ResetSuccessor:**

**while** (**true**)

{

// deal with inputs from resteChannel and inChannel

// use a priSelect

**def** index = alt.priSelect()

**if** (index == 0 )

{

inChannel.read()

**def** resetValue = resetChannel.read()

outChannel.write(resetValue)

}

**else**

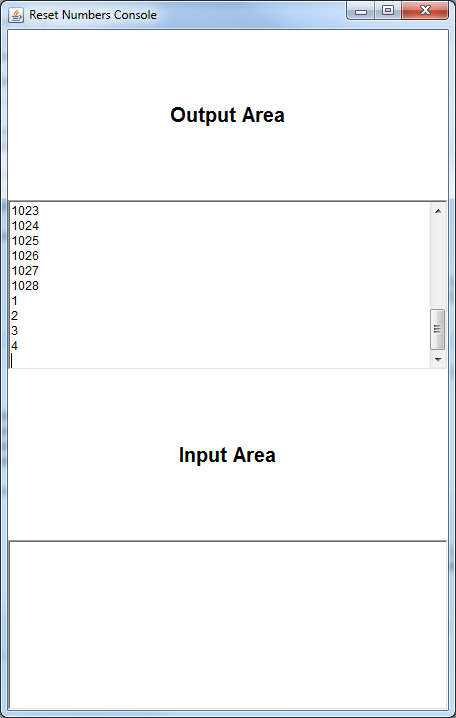
{

outChannel.write(inChannel.read()+1)

}

}

Output:



Questions:

If inchannel.read() is removed from ResetSuccessor the same issue encountered in 4.1 will occur and the program will deadlock.

**Exercise 5.1:**

If the delay is increased for either QProducer or QConsumer the time between the process outputting messages will be equal to that delay. If the delay for both is increased the process will take as long as the greatest delay.

Exercise 5.2:

Controller

Suspend

factor

injector

GPrint

Scale

GFixedDelay

GNumbers

scaledData

timedData

data

Code:

**while** (**true**) {

**switch** ( scaleAlt.priSelect(preCon) ) {

**case** SUSPEND :

// deal with suspend input

**def** inValue = inChannel.read()

**def** result = **new** ScaledData()

result.original = inValue

result.scaled = inValue

outChannel.write ( result )

**break**

**case** INJECT:

// deal with inject input

scaling = injector.read() //this is the resume signal as well

*println* "Injected scaling is $scaling"

suspended = **false**

timeout = timer.read() + DOUBLE\_INTERVAL

timer.setAlarm ( timeout )

**break**

**case** TIMER:

// deal with Timer input

timeout = timer.read() + DOUBLE\_INTERVAL

timer.setAlarm ( timeout )

scaling = scaling \* 2

*println* "Normal Timer: new scaling is ${scaling}"

**break**

**case** INPUT:

**def** inValue = inChannel.read()

**def** result = **new** ScaledData()

result.original = inValue

result.scaled = inValue \* scaling

outChannel.write ( result )

// deal with Input channel

**break**

} //end-switch

} //end-while

Output:

Original Scaled

0 0

1 2

2 4

3 6

4 8

Normal Timer: new scaling is 4

5 20

6 24

7 28

8 32

Normal Timer: new scaling is 8

9 72

10 80

11 88

12 96

13 104

Question:

I find that for this exercise using the preconditions is a more elegant solution as due to there being less conditions to check it will be cheaper to preform.

Exercise 6.1

Diagram:

CreateSetsOfEight

Ex06LTS

GenerateSetsOfThree

connect1

connect2

Code:

Ex06LTS.groovy

**class** Ex06LTS **implements** CSProcess

{

**def** ChannelInput inChannel

**def** ChannelOutput outChannel

**def** debugList = []

**void** run (){

**def** inList = inChannel.read()

**while** (inList[0] != -1)

{

// hint: output list elements as single integers

**for**(j **in** 0 ..< inList.size)

{

outChannel.write(inList[j])

debugList = debugList << inList[j]

}

inList = inChannel.read()

}

outChannel.write(-1)

}

}

Ex06Test.groovy

**class** Ex06Test **extends** GroovyTestCase

{

**void** test()

{

One2OneChannel connect1 = Channel.*createOne2One*()

One2OneChannel connect2 = Channel.*createOne2One*()

**def** GenerateSetsOfThree = **new** GenerateSetsOfThree ( outChannel: connect1.out() )

**def** Ex06LTS = **new** Ex06LTS ( inChannel: connect1.**in**(), outChannel: connect2.out() )

**def** CreateSetsOfEight = **new** CreateSetsOfEight ( inChannel: connect2.**in**() )

**def** processList = [ GenerateSetsOfThree, Ex06LTS, CreateSetsOfEight ]

**new** PAR (processList).run()

**def** expected = Ex06LTS.debugList.subList( Ex06LTS.debugList.size - 8,

Ex06LTS.debugList.size )

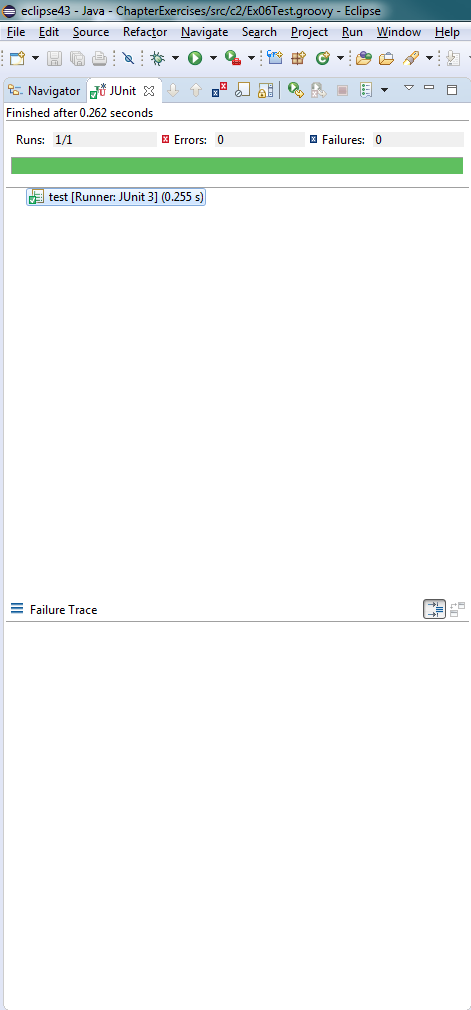
**def** actual = CreateSetsOfEight.outList

*assertTrue*( actual == expected)

}

}

Output:



Exercise 7.1

Diagram:

S02C0send

client0

Client1

server0

Server1

C02S0request

S02S1request

S12S0send

S12S0request

S02S1send

C12S1request

S12C1send

Code:

**switch** (index)

{

**case** CLIENT :

**def** key = clientRequest.read()

*println*("Client: $serverNumber looking for Key: $key in Server: $serverNumber")

**if** ( dataMap.containsKey(key) )

{

clientSend.write(dataMap[key])

*println*("Server: $serverNumber is sending Key: $key to Client: $serverNumber")

}

**else**

thisServerRequest.write(key)

//end if

**break**

**case** OTHER\_REQUEST :

**def** key = otherServerRequest.read()

*println*("Server: " + (1 + (-serverNumber)) + " looking for Key: $key in Server:$serverNumber")

**if** ( dataMap.containsKey(key) )

otherServerSend.write(dataMap[key])

**else**

otherServerSend.write(-1)

//end if

**break**

**case** THIS\_RECEIVE :

clientSend.write(thisServerReceive.read() )

*println*("Server: $serverNumber has recieved value from Server: "+ (1 + (-serverNumber))+"")

**break**

} // end switch

Output:

Client 1 has 10 values in [11, 12, 13, 14, 15, 6, 17, 8, 19, 20]

Client 0 has 10 values in [1, 12, 3, 14, 15, 16, 7, 18, 9, 10]

Client: 0 looking for Key: 1 in Server: 0

Client: 1 looking for Key: 11 in Server: 1

Server: 0 is sending Key: 1 to Client: 0

Server: 1 is sending Key: 11 to Client: 1

Client: 0 looking for Key: 12 in Server: 0

Client: 1 looking for Key: 12 in Server: 1

Server: 1 is sending Key: 12 to Client: 1

Server: 0 looking for Key: 12 in Server: 1

Client: 1 looking for Key: 13 in Server: 1

Server: 1 is sending Key: 13 to Client: 1

Server: 0 has recieved value from Server: 1

Client: 1 looking for Key: 14 in Server: 1

Client: 0 looking for Key: 3 in Server: 0

Server: 1 is sending Key: 14 to Client: 1

Server: 0 is sending Key: 3 to Client: 0

Client: 1 looking for Key: 15 in Server: 1

Client: 0 looking for Key: 14 in Server: 0

Server: 1 is sending Key: 15 to Client: 1

Server: 0 looking for Key: 14 in Server: 1

Client: 1 looking for Key: 6 in Server: 1

Server: 0 has recieved value from Server: 1

Client: 0 looking for Key: 15 in Server: 0

Question:

The deadlock occurs when the servers each try to access each other at the same time. As they are both anticipating a response, neither will be able to progress until the response from the other arrives, which it never will.

Exercise 8.1

Code:

**for** ( i **in** 0 ..< iterations)

{

**def** key = selectList[i]

**def** expected = key\*10

**def** valuesMatch = **false**

requestChannel.write(key)

**def** v = receiveChannel.read()

**if**(v==expected)

valuesMatch = **true**

*println*("Client $clientNumber requested value at location $key. Expected matches Actual:

$valuesMatch")

}

Output:

Client 1 has 10 values in [11, 12, 13, 4, 15, 16, 17, 18, 19, 20]

Client 0 has 10 values in [1, 2, 3, 4, 5, 6, 7, 18, 9, 10]

Client 0 requested value at location 1. Expected matches Actual: true

Client 1 requested value at location 11. Expected matches Actual: true

Client 0 requested value at location 2. Expected matches Actual: true

Client 1 requested value at location 12. Expected matches Actual: true

Client 0 requested value at location 3. Expected matches Actual: true

Client 1 requested value at location 13. Expected matches Actual: true

Client 0 requested value at location 4. Expected matches Actual: true

Client 1 requested value at location 4. Expected matches Actual: true

Client 0 requested value at location 5. Expected matches Actual: true

Client 1 requested value at location 15. Expected matches Actual: true

Client 0 requested value at location 6. Expected matches Actual: true

Client 1 requested value at location 16. Expected matches Actual: true

Client 0 requested value at location 7. Expected matches Actual: true

Client 1 requested value at location 17. Expected matches Actual: true

Client 0 requested value at location 18. Expected matches Actual: true

Client 1 requested value at location 18. Expected matches Actual: true

Client 0 requested value at location 9. Expected matches Actual: true

Client 1 requested value at location 19. Expected matches Actual: true

Client 0 requested value at location 10. Expected matches Actual: true

Client 1 requested value at location 20. Expected matches Actual: true

Client 0 has finished

Client 1 has finished

Exercise 9.1

Diagram:

outChannel

toAccuracy

toBuffer

transfer

inChannel

AccuracyTest

EventPrompter

EventOWBuffer

EventReceiver

get

Code:

**EventHandler.groovy**

**def** handlerList = [ **new** EventReceiver ( eventIn: inChannel,

eventOut: toBuffer.out()),

**new** EventOWBuffer ( inChannel: toBuffer.**in**(),

getChannel: get.**in**(),

outChannel: transfer.out() ),

**new** EventPrompter ( inChannel: transfer.**in**(),

getChannel: get.out(),

outChannel: toAccuracy.out() ),

**new** AccuracyTest (inChannel: toAccuracy.**in**(),

outChannel: outChannel)

]

**EventData.groovy**

**class** EventData **implements** Serializable, JCSPCopy {

**def** **int** source = 0

**def** **int** data = 0

**def** **int** missed = -1

**def** valid = **false**

**def** copy() {

**def** e = **new** EventData ( source: **this**.source,

data: **this**.data,

missed: **this**.missed,

valid: **this**.valid )

**return** e

}

**def** String toString() {

**def** s = "EventData -> [source: "

s = s + source + ", data: "

s = s + data + ", missed: "

s = s + missed + "]"

s = s + " Correct missed: $valid"

**return** s

}

}

**AccuracyTest.groovy**

**class** AccuracyTest **implements** CSProcess

{

**def** ChannelInput inChannel

**def** ChannelOutput outChannel

**void** run()

{

**def** e = inChannel.read()

**def** prev = 0

**while**(**true**)

{

**def** expected = (e.data - prev) - 1

**if**(e.missed == expected)

{

e.valid = **true**

}

prev = e.data

outChannel.write(e)

e = inChannel.read()

}

}

}

Output:

**RunSingleStream.groovy output:**

Event Output

Event Generator for source 1 has started

EventData -> [source: 1, data: 100, missed: 0] Correct missed: false

EventData -> [source: 1, data: 101, missed: 0] Correct missed: true

EventData -> [source: 1, data: 102, missed: 0] Correct missed: true

EventData -> [source: 1, data: 111, missed: 8] Correct missed: true

EventData -> [source: 1, data: 119, missed: 7] Correct missed: true

EventData -> [source: 1, data: 127, missed: 7] Correct missed: true

EventData -> [source: 1, data: 138, missed: 10] Correct missed: true

EventData -> [source: 1, data: 148, missed: 9] Correct missed: true

EventData -> [source: 1, data: 157, missed: 8] Correct missed: true

EventData -> [source: 1, data: 166, missed: 8] Correct missed: true

Source 1 has finished

Exercise 9.2

Data:

|  |  |  |
| --- | --- | --- |
| Test 1 | Min = 10 | Max = 400 |
| Test 2 | Min = 100 | Max = 400 |
| Test 3 | Min = 10 | Max = 1000 |
| Test 4 | Min = 100 | Max = 1000 |
| Test 5 | Min = 100 | Max = 4000 |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test | Source 1 | Source 2 | Source 3 | Source 4 | Source 5 | Source 6 | Source 7 | Source 8 | Source 9 | Total |
| 1 | 93 | 92 | 90 | 94 | 94 | 93 | 94 | 92 | 93 | 835 |
| 2 | 93 | 92 | 91 | 94 | 94 | 95 | 94 | 93 | 93 | 839 |
| 3 | 95 | 94 | 93 | 95 | 95 | 95 | 95 | 94 | 94 | 850 |
| 4 | 95 | 94 | 93 | 95 | 95 | 95 | 96 | 94 | 95 | 852 |
| 5 | 96 | 96 | 96 | 96 | 96 | 94 | 95 | 95 | 95 | 859 |

Output:

**Test 1:**

Event Generator for source 5 has started

Event Output

Event Generator for source 7 has started

Event Generator for source 9 has started

Event Generator for source 2 has started

Event Generator for source 1 has started

Event Generator for source 6 has started

Event Generator for source 3 has started

Event Generator for source 4 has started

Event Generator for source 8 has started

EventData -> [source: 6, data: 600, missed: 0] Correct missed: false

EventData -> [source: 6, data: 601, missed: 0] Correct missed: true

EventData -> [source: 7, data: 700, missed: 0] Correct missed: false

EventData -> [source: 8, data: 800, missed: 0] Correct missed: false

EventData -> [source: 9, data: 900, missed: 0] Correct missed: false

EventData -> [source: 1, data: 100, missed: 0] Correct missed: false

EventData -> [source: 2, data: 200, missed: 0] Correct missed: false

EventData -> [source: 3, data: 300, missed: 0] Correct missed: false

EventData -> [source: 4, data: 400, missed: 0] Correct missed: false

Source 6 has finished

EventData -> [source: 5, data: 500, missed: 0] Correct missed: false

EventData -> [source: 6, data: 602, missed: 0] Correct missed: true

EventData -> [source: 7, data: 701, missed: 0] Correct missed: true

EventData -> [source: 8, data: 801, missed: 0] Correct missed: true

EventData -> [source: 9, data: 901, missed: 0] Correct missed: true

EventData -> [source: 1, data: 101, missed: 0] Correct missed: true

EventData -> [source: 2, data: 201, missed: 0] Correct missed: true

EventData -> [source: 3, data: 301, missed: 0] Correct missed: true

EventData -> [source: 4, data: 401, missed: 0] Correct missed: true

Source 7 has finished

EventData -> [source: 5, data: 501, missed: 0] Correct missed: true

Source 4 has finished

EventData -> [source: 6, data: 603, missed: 0] Correct missed: true

EventData -> [source: 7, data: 704, missed: 2] Correct missed: true

EventData -> [source: 8, data: 803, missed: 1] Correct missed: true

EventData -> [source: 9, data: 909, missed: 7] Correct missed: true

Source 5 has finished

EventData -> [source: 1, data: 112, missed: 10] Correct missed: true

EventData -> [source: 2, data: 207, missed: 5] Correct missed: true

Source 1 has finished

EventData -> [source: 3, data: 308, missed: 6] Correct missed: true

Source 9 has finished

EventData -> [source: 4, data: 428, missed: 26] Correct missed: true

EventData -> [source: 5, data: 526, missed: 24] Correct missed: true

Source 8 has finished

EventData -> [source: 6, data: 689, missed: 85] Correct missed: true

EventData -> [source: 7, data: 750, missed: 45] Correct missed: true

EventData -> [source: 8, data: 835, missed: 31] Correct missed: true

EventData -> [source: 9, data: 939, missed: 29] Correct missed: true

EventData -> [source: 1, data: 147, missed: 34] Correct missed: true

EventData -> [source: 2, data: 231, missed: 23] Correct missed: true

EventData -> [source: 3, data: 323, missed: 14] Correct missed: true

EventData -> [source: 4, data: 471, missed: 42] Correct missed: true

Source 2 has finished

EventData -> [source: 5, data: 564, missed: 37] Correct missed: true

EventData -> [source: 6, data: 698, missed: 8] Correct missed: true

EventData -> [source: 7, data: 798, missed: 47] Correct missed: true

EventData -> [source: 8, data: 869, missed: 33] Correct missed: true

EventData -> [source: 9, data: 972, missed: 32] Correct missed: true

EventData -> [source: 1, data: 183, missed: 35] Correct missed: true

EventData -> [source: 2, data: 256, missed: 24] Correct missed: true

EventData -> [source: 3, data: 347, missed: 23] Correct missed: true

EventData -> [source: 4, data: 498, missed: 26] Correct missed: true

EventData -> [source: 5, data: 598, missed: 33] Correct missed: true

EventData -> [source: 8, data: 897, missed: 27] Correct missed: true

EventData -> [source: 9, data: 998, missed: 25] Correct missed: true

EventData -> [source: 1, data: 198, missed: 14] Correct missed: true

Source 3 has finished

EventData -> [source: 2, data: 277, missed: 20] Correct missed: true

EventData -> [source: 3, data: 364, missed: 16] Correct missed: true

EventData -> [source: 8, data: 898, missed: 0] Correct missed: true

EventData -> [source: 2, data: 298, missed: 20] Correct missed: true

EventData -> [source: 3, data: 384, missed: 19] Correct missed: true

EventData -> [source: 3, data: 396, missed: 11] Correct missed: true

EventData -> [source: 3, data: 398, missed: 1] Correct missed: true

**Test 2:**

Event Generator for source 4 has started

Event Generator for source 8 has started

Event Generator for source 9 has started

Event Generator for source 7 has started

Event Generator for source 3 has started

Event Generator for source 5 has started

Event Generator for source 6 has started

Event Generator for source 2 has started

Event Output

Event Generator for source 1 has started

EventData -> [source: 6, data: 600, missed: 0] Correct missed: false

EventData -> [source: 7, data: 700, missed: 0] Correct missed: false

EventData -> [source: 8, data: 800, missed: 0] Correct missed: false

EventData -> [source: 9, data: 900, missed: 0] Correct missed: false

EventData -> [source: 1, data: 100, missed: 0] Correct missed: false

EventData -> [source: 2, data: 200, missed: 0] Correct missed: false

EventData -> [source: 3, data: 300, missed: 0] Correct missed: false

Source 6 has finished

EventData -> [source: 4, data: 400, missed: 0] Correct missed: false

EventData -> [source: 5, data: 500, missed: 0] Correct missed: false

EventData -> [source: 6, data: 601, missed: 0] Correct missed: true

EventData -> [source: 7, data: 701, missed: 0] Correct missed: true

EventData -> [source: 8, data: 801, missed: 0] Correct missed: true

EventData -> [source: 9, data: 901, missed: 0] Correct missed: true

EventData -> [source: 1, data: 101, missed: 0] Correct missed: true

EventData -> [source: 2, data: 201, missed: 0] Correct missed: true

EventData -> [source: 3, data: 301, missed: 0] Correct missed: true

Source 7 has finished

EventData -> [source: 4, data: 401, missed: 0] Correct missed: true

EventData -> [source: 5, data: 501, missed: 0] Correct missed: true

Source 4 has finished

EventData -> [source: 6, data: 602, missed: 0] Correct missed: true

EventData -> [source: 7, data: 702, missed: 0] Correct missed: true

EventData -> [source: 8, data: 803, missed: 1] Correct missed: true

Source 1 has finished

EventData -> [source: 9, data: 910, missed: 8] Correct missed: true

Source 5 has finished

EventData -> [source: 1, data: 117, missed: 15] Correct missed: true

Source 9 has finished

EventData -> [source: 2, data: 212, missed: 10] Correct missed: true

EventData -> [source: 3, data: 310, missed: 8] Correct missed: true

Source 8 has finished

EventData -> [source: 4, data: 437, missed: 35] Correct missed: true

EventData -> [source: 5, data: 534, missed: 32] Correct missed: true

EventData -> [source: 6, data: 698, missed: 95] Correct missed: true

EventData -> [source: 7, data: 752, missed: 49] Correct missed: true

EventData -> [source: 8, data: 839, missed: 35] Correct missed: true

EventData -> [source: 9, data: 947, missed: 36] Correct missed: true

EventData -> [source: 1, data: 156, missed: 38] Correct missed: true

EventData -> [source: 2, data: 237, missed: 24] Correct missed: true

EventData -> [source: 3, data: 331, missed: 20] Correct missed: true

Source 2 has finished

EventData -> [source: 4, data: 478, missed: 40] Correct missed: true

EventData -> [source: 5, data: 568, missed: 33] Correct missed: true

EventData -> [source: 7, data: 798, missed: 45] Correct missed: true

EventData -> [source: 8, data: 870, missed: 30] Correct missed: true

EventData -> [source: 9, data: 979, missed: 31] Correct missed: true

EventData -> [source: 1, data: 192, missed: 35] Correct missed: true

EventData -> [source: 2, data: 259, missed: 21] Correct missed: true

Source 3 has finished

EventData -> [source: 3, data: 349, missed: 17] Correct missed: true

EventData -> [source: 4, data: 498, missed: 19] Correct missed: true

EventData -> [source: 5, data: 598, missed: 29] Correct missed: true

EventData -> [source: 8, data: 898, missed: 27] Correct missed: true

EventData -> [source: 9, data: 998, missed: 18] Correct missed: true

EventData -> [source: 1, data: 198, missed: 5] Correct missed: true

EventData -> [source: 2, data: 286, missed: 26] Correct missed: true

EventData -> [source: 3, data: 372, missed: 22] Correct missed: true

EventData -> [source: 2, data: 298, missed: 11] Correct missed: true

EventData -> [source: 3, data: 392, missed: 19] Correct missed: true

EventData -> [source: 3, data: 398, missed: 5] Correct missed: true

**Test 3:**

Event Generator for source 9 has started

Event Generator for source 6 has started

Event Output

Event Generator for source 5 has started

Event Generator for source 1 has started

Event Generator for source 4 has started

Event Generator for source 7 has started

Event Generator for source 2 has started

Event Generator for source 3 has started

Event Generator for source 8 has started

EventData -> [source: 6, data: 600, missed: 0] Correct missed: false

EventData -> [source: 7, data: 700, missed: 0] Correct missed: false

EventData -> [source: 8, data: 800, missed: 0] Correct missed: false

EventData -> [source: 9, data: 900, missed: 0] Correct missed: false

Source 6 has finished

EventData -> [source: 1, data: 100, missed: 0] Correct missed: false

EventData -> [source: 2, data: 200, missed: 0] Correct missed: false

EventData -> [source: 3, data: 300, missed: 0] Correct missed: false

Source 7 has finished

EventData -> [source: 4, data: 400, missed: 0] Correct missed: false

EventData -> [source: 5, data: 500, missed: 0] Correct missed: false

Source 4 has finished

EventData -> [source: 6, data: 601, missed: 0] Correct missed: true

Source 1 has finished

EventData -> [source: 7, data: 701, missed: 0] Correct missed: true

Source 5 has finished

Source 9 has finished

EventData -> [source: 8, data: 801, missed: 0] Correct missed: true

EventData -> [source: 9, data: 901, missed: 0] Correct missed: true

Source 8 has finished

EventData -> [source: 1, data: 101, missed: 0] Correct missed: true

EventData -> [source: 2, data: 201, missed: 0] Correct missed: true

Source 2 has finished

EventData -> [source: 3, data: 301, missed: 0] Correct missed: true

EventData -> [source: 4, data: 401, missed: 0] Correct missed: true

EventData -> [source: 5, data: 501, missed: 0] Correct missed: true

EventData -> [source: 6, data: 602, missed: 0] Correct missed: true

EventData -> [source: 7, data: 702, missed: 0] Correct missed: true

EventData -> [source: 8, data: 802, missed: 0] Correct missed: true

EventData -> [source: 9, data: 904, missed: 2] Correct missed: true

Source 3 has finished

EventData -> [source: 1, data: 107, missed: 5] Correct missed: true

EventData -> [source: 2, data: 212, missed: 10] Correct missed: true

EventData -> [source: 3, data: 321, missed: 19] Correct missed: true

EventData -> [source: 4, data: 454, missed: 52] Correct missed: true

EventData -> [source: 5, data: 560, missed: 58] Correct missed: true

EventData -> [source: 6, data: 698, missed: 95] Correct missed: true

EventData -> [source: 7, data: 798, missed: 95] Correct missed: true

EventData -> [source: 8, data: 880, missed: 77] Correct missed: true

EventData -> [source: 9, data: 990, missed: 85] Correct missed: true

EventData -> [source: 1, data: 198, missed: 90] Correct missed: true

EventData -> [source: 2, data: 271, missed: 58] Correct missed: true

EventData -> [source: 3, data: 362, missed: 40] Correct missed: true

EventData -> [source: 4, data: 498, missed: 43] Correct missed: true

EventData -> [source: 5, data: 598, missed: 37] Correct missed: true

EventData -> [source: 8, data: 898, missed: 17] Correct missed: true

EventData -> [source: 9, data: 998, missed: 7] Correct missed: true

EventData -> [source: 2, data: 298, missed: 26] Correct missed: true

EventData -> [source: 3, data: 397, missed: 34] Correct missed: true

EventData -> [source: 3, data: 398, missed: 0] Correct missed: true

**Test 4:**

Event Generator for source 1 has started

Event Generator for source 9 has started

Event Generator for source 4 has started

Event Generator for source 5 has started

Event Generator for source 6 has started

Event Generator for source 3 has started

Event Generator for source 8 has started

Event Generator for source 2 has started

Event Output

Event Generator for source 7 has started

EventData -> [source: 6, data: 600, missed: 0] Correct missed: false

EventData -> [source: 8, data: 800, missed: 0] Correct missed: false

Source 6 has finished

EventData -> [source: 9, data: 900, missed: 0] Correct missed: false

EventData -> [source: 1, data: 100, missed: 0] Correct missed: false

EventData -> [source: 2, data: 200, missed: 0] Correct missed: false

EventData -> [source: 3, data: 300, missed: 0] Correct missed: false

Source 7 has finished

EventData -> [source: 4, data: 400, missed: 0] Correct missed: false

Source 4 has finished

EventData -> [source: 5, data: 500, missed: 0] Correct missed: false

Source 1 has finished

Source 5 has finished

EventData -> [source: 6, data: 601, missed: 0] Correct missed: true

Source 9 has finished

EventData -> [source: 7, data: 700, missed: 0] Correct missed: false

Source 8 has finished

EventData -> [source: 8, data: 801, missed: 0] Correct missed: true

EventData -> [source: 9, data: 901, missed: 0] Correct missed: true

Source 2 has finished

EventData -> [source: 1, data: 101, missed: 0] Correct missed: true

EventData -> [source: 2, data: 201, missed: 0] Correct missed: true

EventData -> [source: 3, data: 301, missed: 0] Correct missed: true

EventData -> [source: 4, data: 401, missed: 0] Correct missed: true

EventData -> [source: 5, data: 501, missed: 0] Correct missed: true

EventData -> [source: 6, data: 602, missed: 0] Correct missed: true

EventData -> [source: 7, data: 701, missed: 0] Correct missed: true

EventData -> [source: 8, data: 802, missed: 0] Correct missed: true

EventData -> [source: 9, data: 909, missed: 7] Correct missed: true

Source 3 has finished

EventData -> [source: 1, data: 126, missed: 24] Correct missed: true

EventData -> [source: 2, data: 230, missed: 28] Correct missed: true

EventData -> [source: 3, data: 317, missed: 15] Correct missed: true

EventData -> [source: 4, data: 456, missed: 54] Correct missed: true

EventData -> [source: 5, data: 560, missed: 58] Correct missed: true

EventData -> [source: 6, data: 698, missed: 95] Correct missed: true

EventData -> [source: 7, data: 798, missed: 96] Correct missed: true

EventData -> [source: 8, data: 888, missed: 85] Correct missed: true

EventData -> [source: 9, data: 998, missed: 88] Correct missed: true

EventData -> [source: 1, data: 198, missed: 71] Correct missed: true

EventData -> [source: 2, data: 293, missed: 62] Correct missed: true

EventData -> [source: 3, data: 365, missed: 47] Correct missed: true

EventData -> [source: 4, data: 498, missed: 41] Correct missed: true

EventData -> [source: 5, data: 598, missed: 37] Correct missed: true

EventData -> [source: 8, data: 898, missed: 9] Correct missed: true

EventData -> [source: 2, data: 298, missed: 4] Correct missed: true

EventData -> [source: 3, data: 397, missed: 31] Correct missed: true

EventData -> [source: 3, data: 398, missed: 0] Correct missed: true

**Test 5:**

Event Generator for source 7 has started

Event Output

Event Generator for source 4 has started

Event Generator for source 6 has started

Event Generator for source 5 has started

Event Generator for source 9 has started

Event Generator for source 3 has started

Event Generator for source 8 has started

Event Generator for source 1 has started

Event Generator for source 2 has started

Source 6 has finished

EventData -> [source: 6, data: 600, missed: 0] Correct missed: false

Source 7 has finished

Source 4 has finished

EventData -> [source: 6, data: 601, missed: 0] Correct missed: true

EventData -> [source: 7, data: 700, missed: 0] Correct missed: false

Source 5 has finished

Source 9 has finished

Source 1 has finished

Source 8 has finished

Source 2 has finished

EventData -> [source: 8, data: 800, missed: 0] Correct missed: false

EventData -> [source: 9, data: 900, missed: 0] Correct missed: false

Source 3 has finished

EventData -> [source: 1, data: 100, missed: 0] Correct missed: false

EventData -> [source: 2, data: 200, missed: 0] Correct missed: false

EventData -> [source: 3, data: 300, missed: 0] Correct missed: false

EventData -> [source: 4, data: 400, missed: 0] Correct missed: false

EventData -> [source: 5, data: 500, missed: 0] Correct missed: false

EventData -> [source: 6, data: 602, missed: 0] Correct missed: true

EventData -> [source: 7, data: 701, missed: 0] Correct missed: true

EventData -> [source: 8, data: 801, missed: 0] Correct missed: true

EventData -> [source: 9, data: 901, missed: 0] Correct missed: true

EventData -> [source: 1, data: 101, missed: 0] Correct missed: true

EventData -> [source: 2, data: 201, missed: 0] Correct missed: true

EventData -> [source: 3, data: 301, missed: 0] Correct missed: true

EventData -> [source: 4, data: 401, missed: 0] Correct missed: true

EventData -> [source: 5, data: 501, missed: 0] Correct missed: true

EventData -> [source: 6, data: 603, missed: 0] Correct missed: true

EventData -> [source: 7, data: 754, missed: 52] Correct missed: true

EventData -> [source: 8, data: 866, missed: 64] Correct missed: true

EventData -> [source: 9, data: 982, missed: 80] Correct missed: true

EventData -> [source: 1, data: 198, missed: 96] Correct missed: true

EventData -> [source: 2, data: 298, missed: 96] Correct missed: true

EventData -> [source: 3, data: 398, missed: 96] Correct missed: true

EventData -> [source: 4, data: 498, missed: 96] Correct missed: true

EventData -> [source: 5, data: 598, missed: 96] Correct missed: true

EventData -> [source: 6, data: 698, missed: 94] Correct missed: true

EventData -> [source: 7, data: 798, missed: 43] Correct missed: true

EventData -> [source: 8, data: 898, missed: 31] Correct missed: true

EventData -> [source: 9, data: 998, missed: 15] Correct missed: true

Questions:

From the experimental results we can see that as the minimum value is increased the total number of misses is increased slightly, while increasing the maximum value will increase the number of misses to a greater effect. However, increasing either value will only have a minor effect on the total number of misses. Despite this it can be seen that more attempts manage to succeed without any misses when the minimum or maximum is raised.

Overall however increasing the maximum or minimum time only seems to increase the number of misses while also increasing the overall runtime, at least while using a fair multiplexer.

Exercise 9.3

Data:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test | Source 1 | Source 2 | Source 3 | Source 4 | Source 5 | Source 6 | Source 7 | Source 8 | Source 9 | Total |
| FairMultiplex | 93 | 92 | 90 | 94 | 94 | 93 | 94 | 92 | 93 | 835 |
| PriMultiplex | 69 | 84 | 83 | 96 | 96 | 94 | 96 | 96 | 96 | 810 |
| Multiplexer | 93 | 92 | 91 | 94 | 94 | 94 | 93 | 93 | 93 | 837 |

Output:

**FairMultiplex**

Event Generator for source 5 has started

Event Output

Event Generator for source 7 has started

Event Generator for source 9 has started

Event Generator for source 2 has started

Event Generator for source 1 has started

Event Generator for source 6 has started

Event Generator for source 3 has started

Event Generator for source 4 has started

Event Generator for source 8 has started

EventData -> [source: 6, data: 600, missed: 0] Correct missed: false

EventData -> [source: 6, data: 601, missed: 0] Correct missed: true

EventData -> [source: 7, data: 700, missed: 0] Correct missed: false

EventData -> [source: 8, data: 800, missed: 0] Correct missed: false

EventData -> [source: 9, data: 900, missed: 0] Correct missed: false

EventData -> [source: 1, data: 100, missed: 0] Correct missed: false

EventData -> [source: 2, data: 200, missed: 0] Correct missed: false

EventData -> [source: 3, data: 300, missed: 0] Correct missed: false

EventData -> [source: 4, data: 400, missed: 0] Correct missed: false

Source 6 has finished

EventData -> [source: 5, data: 500, missed: 0] Correct missed: false

EventData -> [source: 6, data: 602, missed: 0] Correct missed: true

EventData -> [source: 7, data: 701, missed: 0] Correct missed: true

EventData -> [source: 8, data: 801, missed: 0] Correct missed: true

EventData -> [source: 9, data: 901, missed: 0] Correct missed: true

EventData -> [source: 1, data: 101, missed: 0] Correct missed: true

EventData -> [source: 2, data: 201, missed: 0] Correct missed: true

EventData -> [source: 3, data: 301, missed: 0] Correct missed: true

EventData -> [source: 4, data: 401, missed: 0] Correct missed: true

Source 7 has finished

EventData -> [source: 5, data: 501, missed: 0] Correct missed: true

Source 4 has finished

EventData -> [source: 6, data: 603, missed: 0] Correct missed: true

EventData -> [source: 7, data: 704, missed: 2] Correct missed: true

EventData -> [source: 8, data: 803, missed: 1] Correct missed: true

EventData -> [source: 9, data: 909, missed: 7] Correct missed: true

Source 5 has finished

EventData -> [source: 1, data: 112, missed: 10] Correct missed: true

EventData -> [source: 2, data: 207, missed: 5] Correct missed: true

Source 1 has finished

EventData -> [source: 3, data: 308, missed: 6] Correct missed: true

Source 9 has finished

EventData -> [source: 4, data: 428, missed: 26] Correct missed: true

EventData -> [source: 5, data: 526, missed: 24] Correct missed: true

Source 8 has finished

EventData -> [source: 6, data: 689, missed: 85] Correct missed: true

EventData -> [source: 7, data: 750, missed: 45] Correct missed: true

EventData -> [source: 8, data: 835, missed: 31] Correct missed: true

EventData -> [source: 9, data: 939, missed: 29] Correct missed: true

EventData -> [source: 1, data: 147, missed: 34] Correct missed: true

EventData -> [source: 2, data: 231, missed: 23] Correct missed: true

EventData -> [source: 3, data: 323, missed: 14] Correct missed: true

EventData -> [source: 4, data: 471, missed: 42] Correct missed: true

Source 2 has finished

EventData -> [source: 5, data: 564, missed: 37] Correct missed: true

EventData -> [source: 6, data: 698, missed: 8] Correct missed: true

EventData -> [source: 7, data: 798, missed: 47] Correct missed: true

EventData -> [source: 8, data: 869, missed: 33] Correct missed: true

EventData -> [source: 9, data: 972, missed: 32] Correct missed: true

EventData -> [source: 1, data: 183, missed: 35] Correct missed: true

EventData -> [source: 2, data: 256, missed: 24] Correct missed: true

EventData -> [source: 3, data: 347, missed: 23] Correct missed: true

EventData -> [source: 4, data: 498, missed: 26] Correct missed: true

EventData -> [source: 5, data: 598, missed: 33] Correct missed: true

EventData -> [source: 8, data: 897, missed: 27] Correct missed: true

EventData -> [source: 9, data: 998, missed: 25] Correct missed: true

EventData -> [source: 1, data: 198, missed: 14] Correct missed: true

Source 3 has finished

EventData -> [source: 2, data: 277, missed: 20] Correct missed: true

EventData -> [source: 3, data: 364, missed: 16] Correct missed: true

EventData -> [source: 8, data: 898, missed: 0] Correct missed: true

EventData -> [source: 2, data: 298, missed: 20] Correct missed: true

EventData -> [source: 3, data: 384, missed: 19] Correct missed: true

EventData -> [source: 3, data: 396, missed: 11] Correct missed: true

EventData -> [source: 3, data: 398, missed: 1] Correct missed: true

**PriMultiplex**

Event Generator for source 2 has started

Event Output

Event Generator for source 3 has started

Event Generator for source 7 has started

Event Generator for source 1 has started

Event Generator for source 5 has started

Event Generator for source 9 has started

Event Generator for source 6 has started

Event Generator for source 4 has started

Event Generator for source 8 has started

EventData -> [source: 6, data: 600, missed: 0] Correct missed: false

EventData -> [source: 6, data: 601, missed: 0] Correct missed: true

EventData -> [source: 1, data: 100, missed: 0] Correct missed: false

EventData -> [source: 1, data: 101, missed: 0] Correct missed: true

EventData -> [source: 1, data: 102, missed: 0] Correct missed: true

EventData -> [source: 1, data: 109, missed: 6] Correct missed: true

EventData -> [source: 1, data: 110, missed: 0] Correct missed: true

EventData -> [source: 1, data: 114, missed: 3] Correct missed: true

EventData -> [source: 1, data: 121, missed: 6] Correct missed: true

Source 6 has finished

EventData -> [source: 1, data: 123, missed: 1] Correct missed: true

EventData -> [source: 1, data: 124, missed: 0] Correct missed: true

EventData -> [source: 1, data: 125, missed: 0] Correct missed: true

EventData -> [source: 1, data: 129, missed: 3] Correct missed: true

EventData -> [source: 1, data: 135, missed: 5] Correct missed: true

EventData -> [source: 1, data: 138, missed: 2] Correct missed: true

EventData -> [source: 1, data: 144, missed: 5] Correct missed: true

EventData -> [source: 1, data: 149, missed: 4] Correct missed: true

Source 7 has finished

EventData -> [source: 1, data: 154, missed: 4] Correct missed: true

EventData -> [source: 1, data: 155, missed: 0] Correct missed: true

Source 4 has finished

EventData -> [source: 1, data: 157, missed: 1] Correct missed: true

EventData -> [source: 1, data: 161, missed: 3] Correct missed: true

EventData -> [source: 1, data: 168, missed: 6] Correct missed: true

EventData -> [source: 1, data: 173, missed: 4] Correct missed: true

EventData -> [source: 1, data: 178, missed: 4] Correct missed: true

EventData -> [source: 1, data: 183, missed: 4] Correct missed: true

EventData -> [source: 1, data: 184, missed: 0] Correct missed: true

Source 5 has finished

EventData -> [source: 1, data: 188, missed: 3] Correct missed: true

Source 1 has finished

EventData -> [source: 1, data: 189, missed: 0] Correct missed: true

EventData -> [source: 1, data: 190, missed: 0] Correct missed: true

Source 9 has finished

EventData -> [source: 1, data: 193, missed: 2] Correct missed: true

EventData -> [source: 1, data: 196, missed: 2] Correct missed: true

Source 8 has finished

EventData -> [source: 1, data: 198, missed: 1] Correct missed: true

EventData -> [source: 2, data: 200, missed: 0] Correct missed: false

EventData -> [source: 2, data: 201, missed: 0] Correct missed: true

EventData -> [source: 2, data: 271, missed: 69] Correct missed: true

EventData -> [source: 2, data: 272, missed: 0] Correct missed: true

EventData -> [source: 2, data: 276, missed: 3] Correct missed: true

EventData -> [source: 2, data: 277, missed: 0] Correct missed: true

EventData -> [source: 2, data: 278, missed: 0] Correct missed: true

EventData -> [source: 2, data: 283, missed: 4] Correct missed: true

EventData -> [source: 2, data: 284, missed: 0] Correct missed: true

EventData -> [source: 2, data: 287, missed: 2] Correct missed: true

Source 2 has finished

EventData -> [source: 2, data: 289, missed: 1] Correct missed: true

EventData -> [source: 2, data: 292, missed: 2] Correct missed: true

EventData -> [source: 2, data: 293, missed: 0] Correct missed: true

EventData -> [source: 2, data: 295, missed: 1] Correct missed: true

EventData -> [source: 2, data: 298, missed: 2] Correct missed: true

EventData -> [source: 3, data: 300, missed: 0] Correct missed: false

EventData -> [source: 3, data: 301, missed: 0] Correct missed: true

EventData -> [source: 3, data: 373, missed: 71] Correct missed: true

EventData -> [source: 3, data: 376, missed: 2] Correct missed: true

EventData -> [source: 3, data: 378, missed: 1] Correct missed: true

EventData -> [source: 3, data: 382, missed: 3] Correct missed: true

EventData -> [source: 3, data: 385, missed: 2] Correct missed: true

EventData -> [source: 3, data: 387, missed: 1] Correct missed: true

EventData -> [source: 3, data: 388, missed: 0] Correct missed: true

EventData -> [source: 3, data: 389, missed: 0] Correct missed: true

Source 3 has finished

EventData -> [source: 3, data: 391, missed: 1] Correct missed: true

EventData -> [source: 3, data: 394, missed: 2] Correct missed: true

EventData -> [source: 3, data: 395, missed: 0] Correct missed: true

EventData -> [source: 3, data: 396, missed: 0] Correct missed: true

EventData -> [source: 3, data: 397, missed: 0] Correct missed: true

EventData -> [source: 3, data: 398, missed: 0] Correct missed: true

EventData -> [source: 4, data: 400, missed: 0] Correct missed: false

EventData -> [source: 4, data: 401, missed: 0] Correct missed: true

EventData -> [source: 4, data: 498, missed: 96] Correct missed: true

EventData -> [source: 5, data: 500, missed: 0] Correct missed: false

EventData -> [source: 5, data: 501, missed: 0] Correct missed: true

EventData -> [source: 5, data: 598, missed: 96] Correct missed: true

EventData -> [source: 6, data: 602, missed: 0] Correct missed: true

EventData -> [source: 6, data: 603, missed: 0] Correct missed: true

EventData -> [source: 6, data: 698, missed: 94] Correct missed: true

EventData -> [source: 7, data: 700, missed: 0] Correct missed: false

EventData -> [source: 7, data: 701, missed: 0] Correct missed: true

EventData -> [source: 7, data: 798, missed: 96] Correct missed: true

EventData -> [source: 8, data: 800, missed: 0] Correct missed: false

EventData -> [source: 8, data: 801, missed: 0] Correct missed: true

EventData -> [source: 8, data: 898, missed: 96] Correct missed: true

EventData -> [source: 9, data: 900, missed: 0] Correct missed: false

EventData -> [source: 9, data: 901, missed: 0] Correct missed: true

EventData -> [source: 9, data: 998, missed: 96] Correct missed: true

**Multiplexer**

Event Output

Event Generator for source 5 has started

Event Generator for source 2 has started

Event Generator for source 1 has started

Event Generator for source 6 has started

Event Generator for source 4 has started

Event Generator for source 8 has started

Event Generator for source 3 has started

Event Generator for source 9 has started

Event Generator for source 7 has started

EventData -> [source: 6, data: 600, missed: 0] Correct missed: false

EventData -> [source: 6, data: 601, missed: 0] Correct missed: true

EventData -> [source: 7, data: 700, missed: 0] Correct missed: false

EventData -> [source: 8, data: 800, missed: 0] Correct missed: false

EventData -> [source: 9, data: 900, missed: 0] Correct missed: false

EventData -> [source: 1, data: 100, missed: 0] Correct missed: false

Source 6 has finished

EventData -> [source: 2, data: 200, missed: 0] Correct missed: false

EventData -> [source: 3, data: 300, missed: 0] Correct missed: false

EventData -> [source: 4, data: 400, missed: 0] Correct missed: false

EventData -> [source: 5, data: 500, missed: 0] Correct missed: false

EventData -> [source: 6, data: 602, missed: 0] Correct missed: true

EventData -> [source: 7, data: 701, missed: 0] Correct missed: true

EventData -> [source: 8, data: 801, missed: 0] Correct missed: true

EventData -> [source: 9, data: 901, missed: 0] Correct missed: true

EventData -> [source: 1, data: 101, missed: 0] Correct missed: true

EventData -> [source: 2, data: 201, missed: 0] Correct missed: true

EventData -> [source: 3, data: 301, missed: 0] Correct missed: true

EventData -> [source: 4, data: 401, missed: 0] Correct missed: true

EventData -> [source: 5, data: 501, missed: 0] Correct missed: true

Source 7 has finished

EventData -> [source: 6, data: 603, missed: 0] Correct missed: true

EventData -> [source: 7, data: 705, missed: 3] Correct missed: true

Source 4 has finished

EventData -> [source: 8, data: 805, missed: 3] Correct missed: true

EventData -> [source: 9, data: 911, missed: 9] Correct missed: true

EventData -> [source: 1, data: 119, missed: 17] Correct missed: true

Source 1 has finished

Source 5 has finished

EventData -> [source: 2, data: 217, missed: 15] Correct missed: true

EventData -> [source: 3, data: 312, missed: 10] Correct missed: true

Source 9 has finished

EventData -> [source: 4, data: 443, missed: 41] Correct missed: true

EventData -> [source: 5, data: 541, missed: 39] Correct missed: true

Source 8 has finished

EventData -> [source: 6, data: 698, missed: 94] Correct missed: true

EventData -> [source: 7, data: 771, missed: 65] Correct missed: true

EventData -> [source: 8, data: 841, missed: 35] Correct missed: true

EventData -> [source: 9, data: 946, missed: 34] Correct missed: true

EventData -> [source: 1, data: 154, missed: 34] Correct missed: true

EventData -> [source: 2, data: 237, missed: 19] Correct missed: true

EventData -> [source: 3, data: 326, missed: 13] Correct missed: true

EventData -> [source: 4, data: 474, missed: 30] Correct missed: true

Source 2 has finished

EventData -> [source: 5, data: 561, missed: 19] Correct missed: true

EventData -> [source: 7, data: 796, missed: 24] Correct missed: true

EventData -> [source: 8, data: 862, missed: 20] Correct missed: true

EventData -> [source: 9, data: 970, missed: 23] Correct missed: true

EventData -> [source: 1, data: 179, missed: 24] Correct missed: true

EventData -> [source: 2, data: 253, missed: 15] Correct missed: true

EventData -> [source: 3, data: 343, missed: 16] Correct missed: true

EventData -> [source: 4, data: 498, missed: 23] Correct missed: true

EventData -> [source: 5, data: 598, missed: 36] Correct missed: true

Source 3 has finished

EventData -> [source: 7, data: 798, missed: 1] Correct missed: true

EventData -> [source: 8, data: 898, missed: 35] Correct missed: true

EventData -> [source: 9, data: 998, missed: 27] Correct missed: true

EventData -> [source: 1, data: 198, missed: 18] Correct missed: true

EventData -> [source: 2, data: 279, missed: 25] Correct missed: true

EventData -> [source: 3, data: 360, missed: 16] Correct missed: true

EventData -> [source: 2, data: 298, missed: 18] Correct missed: true

EventData -> [source: 3, data: 380, missed: 19] Correct missed: true

EventData -> [source: 3, data: 398, missed: 17] Correct missed: true

Questions:

From the data we can see that while FairMultiplex and multiplexer have little to no discernible difference between them, PriMultiplex has a reduced number of misses overall. In addition to this, the earlier sources produce many less misses than those following them as it selects based on the source index.

Exercise 11.1

Diagram:

data

timedData

Scale

GFixedDelay

GNumbers

scaledData

pause

newScale

oldScale

UIController

Code:

**Scale.groovy**

**while** (**true**) {

**switch** ( normalAlt.priSelect() ) {

**case** NORMAL\_SUSPEND :

suspend.read()

factor.write(scaling.toString()) //Needed as sting for active Label

**def** suspended = **true**

*println* "Suspended"

**while** ( suspended ) {

**switch** ( suspendedAlt.priSelect() ) {

**case** SUSPENDED\_INJECT:

scaling = Integer.*valueOf*(injector.read()) //Needs to be turned back into an int

(from string)

*println* "Injected scaling is $scaling"

suspended = **false**

timeout = timer.read() + DOUBLE\_INTERVAL

timer.setAlarm ( timeout )

**break**

**case** SUSPENDED\_IN:

**def** inValue = inChannel.read()

**def** result = **new** ScaledData()

result.original = inValue

result.scaled = inValue

outChannel.write ( result )

**break**

}

}

**break**

**case** NORMAL\_TIMER:

timeout = timer.read() + DOUBLE\_INTERVAL

timer.setAlarm ( timeout )

scaling = scaling \* multiplier

*println* "Normal Timer: new scaling is $scaling"

**break**

**case** NORMAL\_IN:

**def** inValue = inChannel.read()

**def** result = **new** ScaledData()

result.original = inValue

result.scaled = inValue \* scaling

outChannel.write ( result )

**break**

}

}

**UIController.groovy**

**class** UIController **implements** CSProcess

{

**def** ChannelInput oldScale

**def** ChannelInput data

**def** ChannelOutput pause

**def** ChannelOutput newScale

**void** run()

{

**def** root = **new** ActiveClosingFrame("UIController")

**def** main = root.getActiveFrame()

**def** oldScalelbl = **new** Label("Old scale: ")

**def** scaleValue = **new** ActiveLabel(oldScale)

**def** datatxtbx = **new** ActiveTextArea(data, **null**)

**def** pausebtn = **new** ActiveButton(**null**, pause, "Suspend")

**def** newScalelbl = **new** Label("Insert new scaler:")

**def** newScaletxtbx = **new** ActiveTextEnterField(**null**, newScale)

**def** tempContainer = **new** Container()

tempContainer.setLayout ( **new** GridLayout(1,5) )

tempContainer.add (oldScalelbl)

tempContainer.add (scaleValue)

tempContainer.add (pausebtn)

tempContainer.add (newScalelbl)

tempContainer.add (newScaletxtbx.getActiveTextField())

main.setLayout(**new** BorderLayout())

main.add(datatxtbx, BorderLayout.*CENTER*)

main.add(tempContainer, BorderLayout.*SOUTH*)

main.pack()

main.setVisible(**true**)

**def** network = [root, scaleValue, datatxtbx, pausebtn, newScaletxtbx]

**new** PAR (network).run()

}

}

Output:

