

William Stallings

Operating Systems

Internals and Design Principles

Ninth Edition 2017

Readers/Writers Problem, Ver. 1

INF646 Métodos Formales VK, 2017 - Readers/Writers, v1 1

```
rdr_wrt_msg_v0.pml (1/6)
```

```
$ cat -n rdr_wrt_msg_v0.pml | expand

1  #define NRDRS 5
2  #define NWRTS 2
3  #define MAXRDRQ 20
4  #define MAXWRRQ 20
5
6  chan readrequest = [MAXRDRQ] of { byte, chan }
7  chan writerequest = [MAXWRRQ] of { byte, chan }
8  chan finished = [MAXRDRQ+MAXWRRQ] of { byte }
9  chan mbox[NRDRS+NWRTS+1] = [MAXRDRQ+MAXWRRQ] of { bool }
10
11  byte count = 100
12
```

INF646 Métodos Formales VK, 2017 - Readers/Writers, v1 2

```
rdr_wrt_msg_v0.pml (2/6)
```

```
proctype Reader(byte i) {
13
        readrequest ! i,mbox[i]
14
15
        atomic {
            mbox[i] ?
16
            printf("Reader %d\n",i)
17
18
        finished! i
19
20
21
    proctype Writer(byte i) {
        writerequest ! i,mbox[i]
23
24
        atomic {
25
            mbox[i] ?
            printf("Writer %d\n",i)
26
27
28
        finished! i
29
30
```

```
rdr_wrt_msg_v0.pml (3/6)
```

```
proctype Controller() {
31
32
        byte p
33
34
    end:
35
        do
36
        :: count > 0 ->
37
            if
38
            :: nempty(finished) ->
39
                    atomic {
40
                        finished?p
                        printf("finished %d\n",p)
41
42
43
                    count++
            :: empty(finished) && nempty(writerequest) ->
44
45
                    atomic {
46
                        writerequest ? p
                        printf("request from Writer %d\n",p)
47
48
49
                    count = count - 100
```

3

```
rdr_wrt_msg_v0.pml (4/6)
                                                                                 rdr_wrt_msg_v0.pml (5/6)
                                                                                     71
    50
                  :: empty(finished) && empty(writerequest) && nempty(readrequest) ->
                                                                                              :: count < 0 ->
    51
                           atomic {
                                                                                     72
                                                                                                        atomic {
                                                                                     73
    52
                               readrequest ? p
                                                                                                            finished ? p
                                                                                                            printf("finished Writer %d\n",p)
    53
                               printf("request from Reader %d\n",p)
                                                                                     74
                                                                                     75
    54
    55
                           count - -
                                                                                     76
                                                                                                       count++
                                                                                     77
    56
                           atomic {
                                                                                              od
    57
                               mbox[p] ! true
                                                                                     78
    58
                                                                                     79
                               printf("OK to Reader %d\n",p)
    59
    60
                 fi
    61
             :: count == 0 ->
    62
                      atomic {
    63
                           mbox[p] ! true
                           printf("OK to Writer %d\n",p)
    64
    65
    66
                      atomic {
    67
                           finished ? p
                           printf("finished Writer %d\n",p)
    68
    69
    70
                      count = 100
  INF646 Métodos Formales
                               VK, 2017 - Readers/Writers, v1
                                                                          5
                                                                                    INF646 Métodos Formales
                                                                                                                 VK, 2017 - Readers/Writers, v1
                                                                                                                                                            6
rdr_wrt_msg_v0.pml (6/6)
                                                                                 Simulation: seed 0 (1/2)
                                                                                 $ spin -n0 rdr_wrt_msg_v0.pml | expand
    80
        init {
                                                                                                                            request from Writer 5
    81
             byte i
                                                                                                                            OK to Writer 5
    82
                                                                                                              Writer 5
    83
             atomic {
                                                                                                                            finished Writer 5
    84
                 for (i : 1 .. NRDRS+NWRTS) { /* R1,R2,W3,R4,W5,R6,R7 */
                                                                                                                            request from Writer 3
                                                                                                                            OK to Writer 3
    85
                                                                                                     Writer 3
    86
                      :: i == 3 || i == 5 ->
    87
                               run Writer(i)
                                                                                                                            finished Writer 3
    88
                      :: else ->
                                                                                                                            request from Reader 7
    89
                               run Reader(i)
                                                                                                                            OK to Reader 7
    90
                      fi
                                                                                                                       Reader 7
    91
                                                                                                                            finished 7
                 run Controller()
    92
                                                                                                                            request from Reader 6
                                                                                                                            OK to Reader 6
    93
                                                                                                                   Reader 6
    94 }
                                                                                                                            finished 6
                                                                                                                            request from Reader 4
                                                                                                                            OK to Reader 4
                                                                                                          Reader 4
                                                                                                                            finished 4
  INF646 Métodos Formales
                                VK, 2017 - Readers/Writers, v1
                                                                          7
                                                                                    INF646 Métodos Formales
                                                                                                                 VK, 2017 - Readers/Writers, v1
                                                                                                                                                            8
```

Simulation: seed 0 (2/2)		Simulation: seed 1 (1/2)	Simulation: seed 1 (1/2)	
		\$ spin -n1 rdr_wrt_msg_v0.pml expand		
Reader 2 Reader 1 timeout processes created	request from Reader 2 OK to Reader 2 finished 2 request from Reader 1 OK to Reader 1 finished 1	Writer 3	request from Writer 3 OK to Writer 3 finished Writer 3 request from Writer 5 OK to Writer 5 finished Writer 5 request from Reader 7 OK to Reader 7 Reader 7 finished 7	
seed 0: W5, W3, R7, R6, R4, F	R2, R1.	Reader 2 Reader 4	request from Reader 2 OK to Reader 2 finished 2 request from Reader 4 OK to Reader 4 finished 4	
INF646 Métodos Formales VK, 2017 - F	Readers/Writers, v1	9 INF646 Métodos Formales VK, 20	17 - Readers/Writers, v1	10
Simulation: seed 1 (2/2)		Simulation: seed 2 (1/2)		
Reader 1 timeout 9 processes created	finished 6 request from Reader 1 OK to Reader 1 finished 1	\$ spin -n2 rdr_wrt_msg_v0.pml e Rea Writer Writer 3	request from Reader 6 OK to Reader 6 ader 6 finished 6 request from Writer 5 OK to Writer 5	
seed 0: W5, W3, R7, R6, R4, R5, R6, R4, R6, R4, R6, R4, R6, R4, R6, R6, R4, R6, R6, R4, R6, R6, R6, R6, R6, R6, R6, R6, R6, R6		Reader 2 Reader 1	OK to Reader 2 finished 2 request from Reader 1 OK to Reader 1 finished 1	
INF646 Métodos Formales VK, 2017 - F	Readers/Writers, v1	11 INF646 Métodos Formales VK, 20	17 - Readers/Writers, v1	12

Simulation results Simulation: seed 2 (2/2) request from Reader 7 OK to Reader 7 seed 0: W5, W3, R7, R6, R4, R2, R1. Reader 7 finished 7 seed 1: W3, W5, R7, R2, R4, R6, R1. request from Reader 4 OK to Reader 4 seed 2: R6, W5, W3, R2, R1, R7, R4. Reader 4 finished 4 W3, W5, R4, R7, R1, R2, R6. seed 3: timeout seed 4: W3, W5, R4, R6, R1, R2, R7. 9 processes created W5, W3, R1, R6, (R7 + R2), R4. seed 5: W3, W5, R7, R1, R6, R2, R4. seed 0: W5, W3, R7, R6, R4, R2, R1. seed 6: W3, W5, R7, R2, R4, R6, R1. seed 1: W3, W5, R7, R1, R4, R2, R6. seed 7: R6, W5, W3, R2, R1, R7, R4. R6, W3, W5, R7, R4, R2, R1. seed 2: seed 8: W3, W5, R4, (R1 + R7), (R2 + R6). seed 9: INF646 Métodos Formales VK, 2017 - Readers/Writers, v1 13 INF646 Métodos Formales VK, 2017 - Readers/Writers, v1 14

Observaciones

1. Las estructuras de los códigos de **Reader()** y **Writer()** son idénticas. Se puede crear un solo proctype parametrizado.

Desarrolle el modelo rdr_wrt_msg_v1.pml.

2. ¿Cómo se verifica el nuevo modelo?

Se supone que *Reader* y *Writer* no pueden estar activos a la vez. Tampoco lo pueden 2 *Writers*. Pero sí, pueden estar activos múltiplos *Readers* a la vez.

Modifique el modelo rdr_wrt_msg_v1.pml y verifíquelo.