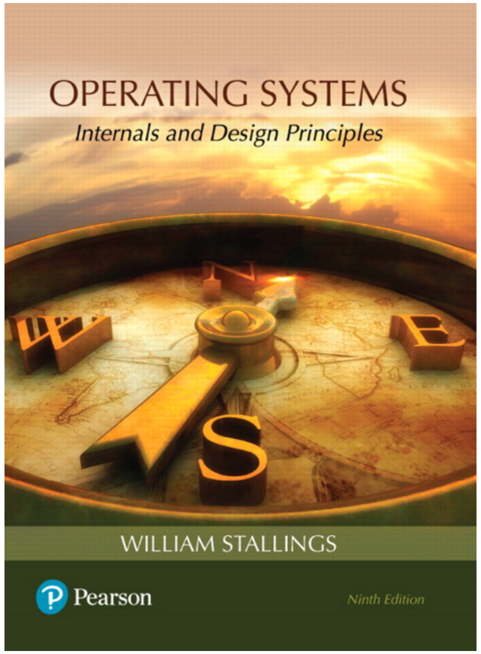


William Stallings

Operating Systems

Internals and Design Principles

Ninth Edition
2017

Readers/Writers Problem, Ver. 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
...
```

rdr_wrt_msg_v0.pml (2/6)

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

rdr_wrt_msg_v0.pml (3/6)

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

rdr_wrt_msg_v0.pml (4/6)

```
...
    ::  empty(finished) && empty(writerequest) && nempty(readrequest) ->
        atomic {
            readrequest ? p
            printf("request from Reader %d\n",p)
        }
        count--
        atomic {
            mbox[p] ! true
            printf("OK to Reader %d\n",p)
        }
    fi
  :: count == 0 ->
      atomic {
          mbox[p] ! true
          printf("OK to Writer %d\n",p)
      }
      atomic {
          finished ? p
          printf("finished Writer %d\n",p)
      }
      count = 100
```

rdr_wrt_msg_v0.pml (5/6)

```
...
  ::  count < 0 ->
        atomic {
            finished ? p
            printf("finished Writer %d\n",p)
        }
        count++
  od
}

...
```

rdr_wrt_msg_v0.pml (6/6)

```
...
init {
    byte i

    atomic {
        for (i : 1 .. NRDRS+NWRTS) {  /* R1,R2,W3,R4,W5,R6,R7 */
            if
            ::  i == 3 || i == 5 ->
                    run Writer(i)
            ::  else ->
                    run Reader(i)
            fi
        }
        run Controller()
    }
}
```

Simulation: seed 0 (1/2)

```
$ spin -n0 rdr_wrt_msg_v0.pml | expand
                   request from Writer 5
                   OK to Writer 5
             Writer 5
                   finished Writer 5
                   request from Writer 3
                   OK to Writer 3
         Writer 3
                   finished Writer 3
                   request from Reader 7
                   OK to Reader 7
                 Reader 7
                   finished 7
                   request from Reader 6
                   OK to Reader 6
               Reader 6
                   finished 6
                   request from Reader 4
                   OK to Reader 4
           Reader 4
                   finished 4
...
```

Simulation: seed 0 (2/2)

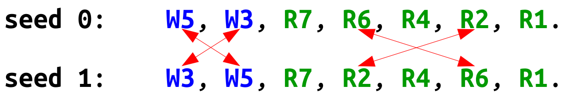
```
...
                                      request from Reader 2
                                      OK to Reader 2
              Reader 2
                                      finished 2
                                      request from Reader 1
                                      OK to Reader 1
          Reader 1
                                      finished 1
      timeout
...
9 processes created



    seed 0:    W5, W3, R7, R6, R4, R2, R1.
```

Simulation: seed 1 (1/2)

```
$ spin -n1 rdr_wrt_msg_v0.pml | expand
                                      request from Writer 3
                                      OK to Writer 3
                  Writer 3
                                      finished Writer 3
                                      request from Writer 5
                                      OK to Writer 5
                          Writer 3
                                      finished Writer 5
                                      request from Reader 7
                                      OK to Reader 7
                                  Reader 7
                                      finished 7
                                      request from Reader 2
                                      OK to Reader 2
              Reader 2
                                      finished 2
                                      request from Reader 4
                                      OK to Reader 4
                      Reader 4
                                      finished 4
...
```

Simulation: seed 1 (2/2)

```
...
                                      request from Reader 6
                                      OK to Reader 6
                              Reader 6
                                      finished 6
                                      request from Reader 1
                                      OK to Reader 1
          Reader 1
                                      finished 1
      timeout
...
9 processes created
```



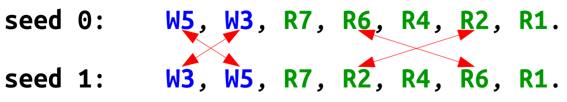
```
    seed 0:    W5, W3, R7, R6, R4, R2, R1.

    seed 1:    W3, W5, R7, R2, R4, R6, R1.
```

Simulation: seed 2 (1/2)

```
$ spin -n2 rdr_wrt_msg_v0.pml | expand
                                      request from Reader 6
                                      OK to Reader 6
                              Reader 6
                                      finished 6
                                      request from Writer 5
                                      OK to Writer 5
                          Writer 5
                                      finished Writer 5
                                      request from Writer 3
                                      OK to Writer 3
                  Writer 3
                                      finished Writer 3
                                      request from Reader 2
                                      OK to Reader 2
              Reader 2
                                      finished 2
                                      request from Reader 1
                                      OK to Reader 1
          Reader 1
                                      finished 1
...
```

Simulation: seed 2 (2/2)

```
...
                                      request from Reader 7
                                      OK to Reader 7
                                  Reader 7
                                      finished 7
                                      request from Reader 4
                                      OK to Reader 4
                      Reader 4
                                      finished 4
      timeout
...
9 processes created
```



```
    seed 0:    W5, W3, R7, R6, R4, R2, R1.

    seed 1:    W3, W5, R7, R2, R4, R6, R1.

    seed 2:    R6, W5, W3, R2, R1, R7, R4.
```

Simulation results

```
seed 0:    W5, W3, R7, R6, R4, R2, R1.

seed 1:    W3, W5, R7, R2, R4, R6, R1.

seed 2:    R6, W5, W3, R2, R1, R7, R4.

seed 3:    W3, W5, R4, R7, R1, R2, R6.

seed 4:    W3, W5, R4, R6, R1, R2, R7.

seed 5:    W5, W3, R1, R6, (R7 + R2), R4.

seed 6:    W3, W5, R7, R1, R6, R2, R4.

seed 7:    W3, W5, R7, R1, R4, R2, R6.

seed 8:    R6, W3, W5, R7, R4, R2, R1.

seed 9:    W3, W5, R4, (R1 + R7), (R2 + R6).
```

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.