Sheet #3

Musab Mehadi

[mmehadi@jacobs-university.de](mailto:mmehadi@jacobs-university.de)

Problem 3.2

a) lets see how the code goes

\*\* let’s assume reader is called and makes mutex=0 and readcount=1 (line 3 and 4)

\*writer =0 (line 5)

\* mutex=1 (line 6)

\*read data and **set mutex to 0** again (line 7 and 8)

\* read count =0

\***set mutex to 1 again(line 10): here the writer will be called but it will not execute as writer =0 but any reader can enter due to mutex=1 creating a loop hole.**

b)this case is same us the first situation but I’ll try to explain it using two readers r1 and r2

\*r1 is called and makes mutex=0 and readcount=1 (line 3 and 4)

\*writer =0 (line 5)

\* mutex=1 (line 6) which means r2 can enter

\*r2 is called and makes mutex=0 again (line 3)

\*readcount=2

\*mutex=2 again (line 6)

\*read data by r1 and **set mutex to 0** (line 7 and 8)

\* read count =1 (line 9)

**\* read data by r2 and down (&mutex ) causes it to wait (line 7 and 8)**

\* the else condition applies to r1(since read count=1) and up(& mutex) is implemented which sets mutex=1.

**\*know r2 can perform the down(&mutex) and readcount =0( lines 8 and 9)**

**\*we enter the if condition and just like question 3.2 a the writer will not be able to enter the critical section.**

\***set mutex to 1 again(line 10): here the writer will be called but it will not execute as writer =0 but any reader can enter due to mutex=1 creating a loop hole.**

c)in this situation

\* let’s assume reader is called and makes mutex=0 and readcount=1 (line 3 and 4)

\*writer =0 (line 5)

\* mutex=1 (line 6)

\*read data and **set mutex to 0** again (line 7 and 8)

\* read count =0

\*writer =1 (line 10)

**\* writer process will be called**

\*writer =0 (line 3)

**\*here (line 4) there will be a loophole when we call down mutex as mutex value was 0 and performing down( &mutex ) will make it wait and hence both reader and writer will not proceed.**