Concurrent and Distributed Systems

Laboratory Assignment 2

November 17, 2019

Student: Marcu Andrei Cristian

Computers and Information Technology

CEN 3.2 A

3rd Year

Problems statements

Problem 1 (50p)
Implement Dekker algorithm using 2 processes (as available in Critical Sections Laboratories). Assignment: Implement the Dekker algorithm in Promela, describe the output and different scenarios (in multiple executions)

Problem 2 (50p)
What values can n have at the end of the concurrent counting algorithm execution? Assignment: Implement this algorithm in Promela. What values do you get at the end for n? Assignment: Prove the above by describing in the technical reports a few test scenarios and different states as shown in the first course. Assignment: Describe the output and different scenarios (in multiple executions).

Concurrent counting	
integer n ← 0	
p	q
integer temp	integer temp
p1: do 10 times	q1: do 10 times
p2: temp ← n	q2: temp ← n
p3: n ← temp + 1	q3: n ← temp +1

Problem1: Dekker algorithm is designed to solve the critical section problem. It synchronizes two threads to enter one by one in a critical section without conflict. I've used the version presented at the course. It works as follows:
We have two flags : wantq and wantp. When wantq is 1 the thread q wants to enter in the critical section, same for wantp.
We have an integer turn which can have the value 1 or 2. It saves the priority in case if both threads want to enter the critical section at the same time.

Problem2: As we've seen in the statement we have 2 threads incrementing the same variable n 10 times. N is a global variable so it will have the same value for all threads.
For example the result for the input presented in the solution (2 threads, 10 iterations each) n will reach a random value close to 20 or exactly 20. (20 is the correct answer).
I'll explain more examples later on my presentation.
Now that we've clarified the Problems Statements we can move on to the implementation.

Solution Problem 1

```
//--------------------------dekker.pml--------------------------
bool wantp = false, wantq = false;
byte turn = 1;

active proctype p() {
	do
	:: wantp = true;
		do
		:: wantq ->
			if
			:: (turn == 2) ->
				wantp = false; turn == 1; wantp = true
			:: else
			fi
		:: else -> break
		od;
		printf("P se afla in sectiunea critica\n");
		turn = 2;
		wantp = false
	od
}
active proctype q() {
	do
	:: wantq = true;
		do
		:: wantp ->
			if
			:: (turn == 1) ->
				wantq = false; turn == 2; wantq = true
			:: else
			fi
		:: else -> break
		od;
		printf("Q se afla in sectiunea critica\n");
		turn = 1;
		wantq = false
	od
}
```

Experiments and results for problem 1

Even though it's an infinite loop all examples will have 250 steps. 1. Random example 1

```
1 q      33  printf('Q se a 2          0          1
1 q      34  turn = 1       2          0          1
1 q      35  wantq = 0      1          0          1
0 p      5   wantp = 1      1          0          0
0 p      7   else           1          1          0
P se afla in sectiunea critica
0 p      16  printf('P se a 1          1          0
0 p      17  turn = 2       1          1          0
0 p      18  wantp = 0      2          1          0
1 q      22  wantq = 1      2          0          0
1 q      24  else           2          0          1
Q se afla in sectiunea critica
1 q      33  printf('Q se a 2          0          1
1 q      34  turn = 1       2          0          1
1 q      35  wantq = 0      1          0          1
Process Statement           turn       wantp      wantq
0 p      5   wantp = 1      1          0          0
1 q      22  wantq = 1      1          1          0
0 p      7   wantq          1          1          1
0 p      9   else           1          1          1
1 q      24  wantp          1          1          1
0 p      7   wantq          1          1          1
0 p      9   else           1          1          1
0 p      7   wantq          1          1          1
0 p      9   else           1          1          1
1 q      26  turn==1        1          1          1
0 p      7   wantq          1          1          1
0 p      9   else           1          1          1
0 p      7   wantq          1          1          1
1 q      28  wantq = 0      1          1          1
0 p      9   else           1          1          0
0 p      7   else           1          1          0
P se afla in sectiunea critica
0 p      16  printf('P se a 1          1          0
0 p      17  turn = 2       1          1          0
1 q      28  turn==2        2          1          0
1 q      28  wantq = 1      2          1          0
Process Statement           turn       wantp      wantq
0 p      18  wantp = 0      2          1          1
0 p      5   wantp = 1      2          0          1
1 q      24  wantp          2          1          1
1 q      26  else           2          1          1
0 p      7   wantq          2          1          1
0 p      9   turn==2        2          1          1
1 q      24  wantp          2          1          1
0 p      11  wantp = 0      2          1          1
1 q      26  else           2          0          1
-------------
depth-limit (-u250 steps) reached
#processes: 2
250:    proc  1 (q) dekker.pml:24 (state 12)
250:    proc  0 (p) dekker.pml:11 (state 5)
2 processes created
```

2. Random example 2

```
1 q      28  wantq = 1       2           1           0
1 q      24  wantp           2           1           1
1 q      26  else            2           1           1
P se afla in sectiunea critica
0 p      16  printf('P se a 2           1           1
0 p      17  turn = 2        2           1           1
0 p      18  wantp = 0       2           1           1
0 p      5   wantp = 1       2           0           1
1 q      24  wantp           2           1           1
1 q      26  else            2           1           1
0 p      7   wantq           2           1           1
0 p      9   turn==2         2           1           1
Process Statement            turn        wantp       wantq
0 p      11  wantp = 0       2           1           1
1 q      24  else            2           0           1
Q se afla in sectiunea critica
1 q      33  printf('Q se a 2           0           1
1 q      34  turn = 1        2           0           1
0 p      11  turn==1         1           0           1
1 q      35  wantq = 0       1           0           1
1 q      22  wantq = 1       1           0           0
0 p      11  wantp = 1       1           0           1
1 q      24  wantp           1           1           1
1 q      26  turn==1         1           1           1
1 q      28  wantq = 0       1           1           1
0 p      7   else            1           1           0
P se afla in sectiunea critica
0 p      16  printf('P se a 1           1           0
0 p      17  turn = 2        1           1           0
1 q      28  turn==2         2           1           0
0 p      18  wantp = 0       2           1           0
0 p      5   wantp = 1       2           0           0
1 q      28  wantq = 1       2           1           0
0 p      7   wantq           2           1           1
0 p      9   turn==2         2           1           1
Process Statement            turn        wantp       wantq
0 p      11  wantp = 0       2           1           1
1 q      24  else            2           0           1
Q se afla in sectiunea critica
1 q      33  printf('Q se a 2           0           1
1 q      34  turn = 1        2           0           1
1 q      35  wantq = 0       1           0           1
0 p      11  turn==1         1           0           0
0 p      11  wantp = 1       1           0           0
0 p      7   else            1           1           0
P se afla in sectiunea critica
0 p      16  printf('P se a 1           1           0
-------------
depth-limit (-u250 steps) reached
#processes: 2
250:    proc  1 (q) dekker.pml:22 (state 18)
250:    proc  0 (p) dekker.pml:17 (state 16)
2 processes created
```

3. Random example 3

```
0 p      17  turn = 2        1          1          0
1 q      28  turn==2         2          1          0
1 q      28  wantq = 1       2          1          0
0 p      18  wantp = 0       2          1          1
1 q      24  else            2          0          1
0 p      5   wantp = 1       2          0          1
Q se afla in sectiunea critica
1 q      33  printf('Q se a 2          1          1
0 p      7   wantq           2          1          1
0 p      9   turn==2         2          1          1
1 q      34  turn = 1        2          1          1
1 q      35  wantq = 0       1          1          1
1 q      22  wantq = 1       1          1          0
0 p      11  wantp = 0       1          1          1
0 p      11  turn==1         1          0          1
0 p      11  wantp = 1       1          0          1
Process Statement           turn       wantp      wantq
1 q      24  wantp           1          1          1
1 q      26  turn==1         1          1          1
1 q      28  wantq = 0       1          1          1
0 p      7   else            1          1          0
P se afla in sectiunea critica
0 p      16  printf('P se a 1          1          0
0 p      17  turn = 2        1          1          0
0 p      18  wantp = 0       2          1          0
0 p      5   wantp = 1       2          0          0
0 p      7   else            2          1          0
P se afla in sectiunea critica
0 p      16  printf('P se a 2          1          0
0 p      17  turn = 2        2          1          0
1 q      28  turn==2         2          1          0
1 q      28  wantq = 1       2          1          0
0 p      18  wantp = 0       2          1          1
1 q      24  else            2          0          1
0 p      5   wantp = 1       2          0          1
0 p      7   wantq           2          1          1
0 p      9   turn==2         2          1          1
Q se afla in sectiunea critica
1 q      33  printf('Q se a 2          1          1
1 q      34  turn = 1        2          1          1
Process Statement           turn       wantp      wantq
1 q      35  wantq = 0       1          1          1
0 p      11  wantp = 0       1          1          0
0 p      11  turn==1         1          0          0
1 q      22  wantq = 1       1          0          0
1 q      24  else            1          0          1
0 p      11  wantp = 1       1          0          1
-------------
depth-limit (-u250 steps) reached
#processes: 2
250:    proc  1 (q) dekker.pml:33 (state 15)
250:    proc  0 (p) dekker.pml:16 (state 13)
2 processes created
```

In the examples presented before we can see only the end of the execution. Full examples will be found in the Experimental data and results folder
As we can see in all examples presented before, the threads enter the critical section one by one when the flag is true.

Conclusions Problem 1

Working on this problem, I have acquired lots of knowledge about thread programming, about the problems that can appear when we are using threads. Dekker algorithm is one good solution to avoid critical section problem and make two threads work with the same variable without modifiying it in the same time. The issue that Dekker algorithm avoids is presented in the second problem.

Solution Problem 2

```
//---------------------------concurrent.pml---------------------------

int n = 0;

active[2] proctype pq() {
        int i = 0, temp;
        do
        :: i < 10 ->
                        temp = n;
                        n = temp + 1;
                        i = i + 1;
        :: else -> break;
        od;
        printf("n = %d \n", n);
}
```

Experiments and results for problem 2

1. Example with 10 iterations.

```
0 pq    5   i<10           6          4          4          5          6
0 pq    7   temp = n       6          4          4          5          6
1 pq    8   n = (temp+1)   6          4          6          5          6
1 pq    9   i = (i+1)      7          4          6          5          6
0 pq    8   n = (temp+1)   7          4          6          6          6
0 pq    9   i = (i+1)      7          4          6          6          6
1 pq    5   i<10           7          5          6          6          6
1 pq    7   temp = n       7          5          6          6          6
1 pq    8   n = (temp+1)   7          5          6          6          7
0 pq    5   i<10           8          5          6          6          7
Process Statement          n          pq(0):i    pq(0):temp pq(1):i    pq(1):temp
1 pq    9   i = (i+1)      8          5          6          6          7
0 pq    7   temp = n       8          5          6          7          7
0 pq    8   n = (temp+1)   8          5          8          7          7
1 pq    5   i<10           9          5          8          7          7
0 pq    9   i = (i+1)      9          5          8          7          7
1 pq    7   temp = n       9          6          8          7          7
1 pq    8   n = (temp+1)   9          6          8          7          9
0 pq    5   i<10           10         6          8          7          9
0 pq    7   temp = n       10         6          8          7          9
1 pq    9   i = (i+1)      10         6          10         7          9
0 pq    8   n = (temp+1)   10         6          10         8          9
1 pq    5   i<10           11         6          10         8          9
1 pq    7   temp = n       11         6          10         8          9
0 pq    9   i = (i+1)      11         6          10         8          11
1 pq    8   n = (temp+1)   11         7          10         8          11
0 pq    5   i<10           12         7          10         8          11
1 pq    9   i = (i+1)      12         7          10         8          11
1 pq    5   i<10           12         7          10         9          11
0 pq    7   temp = n       12         7          10         9          11
0 pq    8   n = (temp+1)   12         7          12         9          11
Process Statement          n          pq(0):i    pq(0):temp pq(1):i    pq(1):temp
0 pq    9   i = (i+1)      13         7          12         9          11
0 pq    5   i<10           13         8          12         9          11
0 pq    7   temp = n       13         8          12         9          11
0 pq    8   n = (temp+1)   13         8          13         9          11
1 pq    7   temp = n       14         8          13         9          11
0 pq    9   i = (i+1)      14         8          13         9          14
1 pq    8   n = (temp+1)   14         9          13         9          14
1 pq    9   i = (i+1)      15         9          13         9          14
0 pq    5   i<10           15         9          13         10         14
0 pq    7   temp = n       15         9          13         10         14
0 pq    8   n = (temp+1)   15         9          15         10         14
0 pq    9   i = (i+1)      16         9          15         10         14
1 pq    5   else           16         10         15         10         14
0 pq    5   else           16         10         15         10         14
n = 16
0 pq    12  printf('n = %d 16         10         15         10         14
n = 16
1 pq    12  printf('n = %d 16         10         15         10         14
106:    proc  1 (pq) terminates
106:    proc  0 (pq) terminates
2 processes created
```

2. Example with 10 iterations.

```
1 pq    9   i = (i+1)      9           5           8           3           6
1 pq    5   i<10           9           5           8           4           6
0 pq    9   i = (i+1)      9           5           8           4           6
1 pq    7   temp = n       9           6           8           4           6
1 pq    8   n = (temp+1)   9           6           8           4           9
1 pq    9   i = (i+1)      10          6           8           4           9
0 pq    5   i<10           10          6           8           5           9
0 pq    7   temp = n       10          6           8           5           9
0 pq    8   n = (temp+1)   10          6           10          5           9
0 pq    9   i = (i+1)      11          6           10          5           9
1 pq    5   i<10           11          7           10          5           9
1 pq    7   temp = n       11          7           10          5           9
1 pq    8   n = (temp+1)   11          7           10          5           11
Process Statement          n           pq(0):i     pq(0):temp pq(1):i     pq(1):temp
1 pq    9   i = (i+1)      12          7           10          5           11
1 pq    5   i<10           12          7           10          6           11
1 pq    7   temp = n       12          7           10          6           11
0 pq    5   i<10           12          7           10          6           12
1 pq    8   n = (temp+1)   12          7           10          6           12
0 pq    7   temp = n       13          7           10          6           12
1 pq    9   i = (i+1)      13          7           13          6           12
0 pq    8   n = (temp+1)   13          7           13          7           12
0 pq    9   i = (i+1)      14          7           13          7           12
1 pq    5   i<10           14          8           13          7           12
1 pq    7   temp = n       14          8           13          7           12
1 pq    8   n = (temp+1)   14          8           13          7           14
0 pq    5   i<10           15          8           13          7           14
0 pq    7   temp = n       15          8           13          7           14
0 pq    8   n = (temp+1)   15          8           15          7           14
0 pq    9   i = (i+1)      16          8           15          7           14
0 pq    5   i<10           16          9           15          7           14
1 pq    9   i = (i+1)      16          9           15          7           14
1 pq    5   i<10           16          9           15          8           14
0 pq    7   temp = n       16          9           15          8           14
Process Statement          n           pq(0):i     pq(0):temp pq(1):i     pq(1):temp
1 pq    7   temp = n       16          9           16          8           14
1 pq    8   n = (temp+1)   16          9           16          8           16
1 pq    9   i = (i+1)      17          9           16          8           16
1 pq    5   i<10           17          9           16          9           16
1 pq    7   temp = n       17          9           16          9           16
0 pq    8   n = (temp+1)   17          9           16          9           17
0 pq    9   i = (i+1)      17          9           16          9           17
1 pq    8   n = (temp+1)   17          10          16          9           17
0 pq    5   else           18          10          16          9           17
1 pq    9   i = (i+1)      18          10          16          9           17
1 pq    5   else           18          10          16          10          17
n = 18
1 pq    12  printf('n = %d 18          10          16          10          17
104:    proc  1 (pq) terminates
n = 18
0 pq    12  printf('n = %d 18          10          16          10          17
106:    proc  0 (pq) terminates
2 processes created
```

3. Example with 10 iterations.

```
1 pq    5   i<10           8          4          8          5          7
0 pq    8   n = (temp+1)   8          4          8          5          7
0 pq    9   i = (i+1)      9          4          8          5          7
1 pq    7   temp = n       9          5          8          5          7
1 pq    8   n = (temp+1)   9          5          8          5          9
1 pq    9   i = (i+1)      10         5          8          5          9
0 pq    5   i<10           10         5          8          6          9
1 pq    5   i<10           10         5          8          6          9
0 pq    7   temp = n       10         5          8          6          9
0 pq    8   n = (temp+1)   10         5          10         6          9
Process Statement          n          pq(0):i    pq(0):temp pq(1):i    pq(1):temp
0 pq    9   i = (i+1)      11         5          10         6          9
1 pq    7   temp = n       11         6          10         6          9
1 pq    8   n = (temp+1)   11         6          10         6          11
1 pq    9   i = (i+1)      12         6          10         6          11
0 pq    5   i<10           12         6          10         7          11
0 pq    7   temp = n       12         6          10         7          11
0 pq    8   n = (temp+1)   12         6          12         7          11
0 pq    9   i = (i+1)      13         6          12         7          11
1 pq    5   i<10           13         7          12         7          11
1 pq    7   temp = n       13         7          12         7          11
0 pq    5   i<10           13         7          12         7          13
1 pq    8   n = (temp+1)   13         7          12         7          13
0 pq    7   temp = n       14         7          12         7          13
0 pq    8   n = (temp+1)   14         7          14         7          13
0 pq    9   i = (i+1)      15         7          14         7          13
1 pq    9   i = (i+1)      15         8          14         7          13
1 pq    5   i<10           15         8          14         8          13
0 pq    5   i<10           15         8          14         8          13
0 pq    7   temp = n       15         8          14         8          13
0 pq    8   n = (temp+1)   15         8          15         8          13
Process Statement          n          pq(0):i    pq(0):temp pq(1):i    pq(1):temp
0 pq    9   i = (i+1)      16         8          15         8          13
1 pq    7   temp = n       16         9          15         8          13
1 pq    8   n = (temp+1)   16         9          15         8          16
1 pq    9   i = (i+1)      17         9          15         8          16
0 pq    5   i<10           17         9          15         9          16
1 pq    5   i<10           17         9          15         9          16
0 pq    7   temp = n       17         9          15         9          16
0 pq    8   n = (temp+1)   17         9          17         9          16
0 pq    9   i = (i+1)      18         9          17         9          16
1 pq    7   temp = n       18         10         17         9          16
0 pq    5   else           18         10         17         9          18
1 pq    8   n = (temp+1)   18         10         17         9          18
n = 19
0 pq    12  printf('n = %d 19         10         17         9          18
1 pq    9   i = (i+1)      19         10         17         9          18
1 pq    5   else           19         10         17         10         18
n = 19
1 pq    12  printf('n = %d 19         10         17         10         18
106:    proc  1 (pq) terminates
106:    proc  0 (pq) terminates
2 processes created
```

Conclusions Problem 2

The conclusion of the examples presented before is that n should be 2 * number of iterations. Well this isn't the value that we print. This happens because the threads are accesing the same value at the same time and it skips incrementations. The part where we increment the value of n it's the critical section of the problem. To avoid this we can use an algorithm like the one presented before, Dekker's algorithm.

References

https://www.geeksforgeeks.org/dekkers-algorithm-in-process-synchronization/