INFO-F-404 : Operating Systems II

1 Exercises

Exercise 1: Let's consider the following protocol of management of critical sections.

First try	
integer turn $\leftarrow$ 1	
p	q
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: await turn = 1	q2: await turn = 2
p3: critical section	q3: critical section
p4: turn $\leftarrow$ 2	q4: turn $\leftarrow$ 1

a) Show that this protocol can not be used in order to manage mutex sections. Use this simplified version of the protocol in order to minimize the size of diagrams.

First try (simplified)	
integer turn $\leftarrow$ 1	
p	q
loop forever	loop forever
p1: await turn = 1	q1: await turn = 2
p2: turn $\leftarrow$ 2	q2: turn $\leftarrow$ 1

b) Use the state diagram in order to show that this protocol guarantees the absence of deadlock.

c) Show that this first protocol could lead to livelock (if the time that one process passes in the critical section is not limited).

Exercise 2: Let's consider the following protocol of management of critical sections.

Second try	
boolean wantp ← false, wantq ← false	
p	q
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: await wantq = false	q2: await wantp = false
p3: wantp ← true	q3: wantq ← true
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

a) Show that this protocol can not guarantee mutual exclusion.

Second try (simplified)	
boolean wantp ← false, wantq ← false	
p	q
loop forever	loop forever
p1: await wantq = false	q1: await wantp = false
p2: wantp ← true	q2: wantq ← true
p3: wantp ← false	q3: wantq ← false

b) Could we have a deadlock and/or livelock ?

Exercise 3: Let's consider the following protocol of management of critical sections.

Third try	
boolean wantp ← false, wantq ← false	
p	q
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

a) Write a simplified version of this protocol.

b) Create a state diagram of this protocol and verify that we have mutual exlusion.

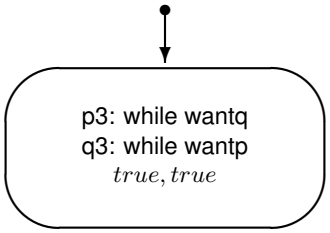
c) Show that this protocol could lead to a deadlock.

d) Describe a scenario that can lead to a deadlock.

Exercise 4: Let's consider the following protocol of management of critical sections. This protocol guarantees the absence of deadlocks and mutual exclusion.

Fourth try	
boolean wantp ← false, wantq ← false	
p	q
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: while wantq	q3: while wantp
p4: wantp ← false	q4: wantq ← false
p5: wantp ← true	q5: wantq ← true
p6: critical section	q6: critical section
p7: wantp ← false	q7: wantq ← false

a) Use state diagram and show that this protocol could lead to livelock.



b) Why is it a livelock (and not a deadlock) ?

Exercise 5: Consider Lamport's algorithm for mutex sections.

Bakery algorithm (2 proces)	
integer np ← 0, nq ← 0	
p	q
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: np ← nq + 1	q2: nq ← np + 1
p3: await nq = 0 or np ≤ nq	q3: await np = 0 or nq < np
p4: critical section	q4: critical section
p5: np ← 0	q5: nq ← 0

a) Use a state diagram in order to show that this algorithm is correct.