

TI-TEDI Exercise: Round-robin token passing algorithm for mutual exclusion

Contents

Introduction.....	1
Exercise A – Implement algorithm in BACI.....	2
Exercise B – Model the algorithm in Spin.....	2

Introduction

In this exercise we are going to investigate the “Round-robin token passing” algorithm for mutual exclusion. As presented in the slides, the algorithm can be summarized like this:

- The token is simply circulated continuously between all processes
- A process that wants to enter the CR, waits until the token comes around, then enters the CR
- When a process exits its CR, or if it don't need to enter its CR, it sends the token on to the next process

Global_initialization:

- Create processes and connect them in a Circle (circular graph)
- Create 1 token and send it one process

/* Performed by each process at start up */

Process_init:

 need_token = false

/* Performed each time a process wants to enter CR */

Process_Enter_CR()

 need_token = true

 wait until token is received

 /* critical region */

 need_token = false

 send tokenmsg to next process

/* Performed each time a process receives a token message */

Process_Handle_ReceiveTokenMsg()

 if (!need_token)

 send tokenmsg to next process

Exercise A – Implement algorithm in BACI

You must implement the algorithm in BACI and demonstrate that it works for at least 3 processes.

You will need to implement channels for inter-process communication somehow. You can either do it yourself, or use the file “channel.cm” you can find together with this exercise. It implements a simple uni-directional channel mechanism. The file “readerwriter.cm”, also found together with this exercise, shows an example of how the channels are used.

Exercise B – Model the algorithm in Spin

You must implement the algorithm in Spin and demonstrate that it works for at least 3 processes. You should use your model to demonstrate that the algorithm is free of deadlock and starvation.