UNIVERSITY
OF SKÖVDE

School of

WRITTEN EXAMINATION

Course: Concurrent Programming

Examination

Course code: IT404G

Credits for written examination: 4.5

Date: 2025-03-07

Examination time: 5 hours

Examination responsible: Richard Senington

Teachers concerned: Birgitta Lindström

Aid at the exam/appendices

Other

Instructions

- ☐ Take a new sheet of paper for each teacher.
- ☐ Take a new sheet of paper when starting a new question.
- ☒ Write only on one side of the paper.
- ☒ Write your name and personal ID No. on all pages you hand in.
- ☒ Use page numbering.
- ☒ Don´t use a red pen.
- ☒ Mark answered questions with a cross on the cover sheet.

Grade points

Examination results should be made public within 18 working days

Good luck!

Total number of pages

Grading

There are five main questions in the written exam corresponding to the course objectives. Each main question consists of a set of three sub-questions, which are graded pass or fail. To pass the exam, you need to ***pass at least one sub-question for each of the main questions****. The more sub-questions you pass, the higher your grade will be. The detailed grading scheme is published in Canvas. For your convenience, each section lists the relevant examination criterion.*

Main question 1

Examination criterion: *Redogöra för olika frågor som måste hanteras i program med samtidigt exekverande processer, inklusive tävlan om resurser och ömsesidig uteslutning*

Sub-question 1a

Does the following code fulfill all three safety properties: mutual exclusion, absence of deadlock and absence of unnecessary delay. You should assume that the semaphore is a blocking FIFO-semaphore.

Sem s(1) Int counter=0		
Process 1	Process 2	Process 3

Int x=0;
While true{
  P(s)
  counter++
  x=counter
  Print(x)
  V(s)
}
	Int x=0;
	While true{
	  P(s)
	  x=counter+2
	  Counter=x
	  V(s)
	}
		Int x=0;
		While true{
		  P(s)
		  counter+=3
		  counter/=2
		  x=counter
		  V(s)
		  Print(x)
		}

Sub-question 1b

Describe the difference between unnecessary delay and deadlock.

Sub-question 1c

Describe resource contention and the concept of mutual exclusion. Explain how they are related to each other.

Main question 2

Examination criterion: *Identifiera, beskriva och diskutera klassiska synkroniseringsproblem mellan parallella processer såsom synkronisering av läsare och skrivare eller av producenter och konsumenter*

Sub-question 2a

Please identify and describe the classic problem seen in the following description. Please identify different processes, what these processes are doing, and what the synchronization and communication mechanisms might be. For the mechanism you choose please explain how it will work and why you chose it.

At a meeting one person is tasked with taking the minutes of the meeting on a computer. Anyone can ask to look at the minutes at any time, and several people can look at the screen at once, before giving the computer back.

Sub-question 2b

Please identify and describe the classic problem seen in the following code. Please identify what each process is doing. Please identify any problems that you can see. Message passing is a-synchronous.

channel c1, c2;				
Process 1	Process 2	Process 3	Process 4	Process 5

While True{
  m=make_next()
  c1.send(m)
}
	While True{
	  c1.receive(n)
	  transform(n)
	  c2.send(n)
	}
		While True{
		  c1.receive(o)
		  transform(o)
		  c2.send(o)
		}
			While True{
			  c2.receive(p)
			  use(p)
			}
				While True{
				  c2.receive(p)
				  use(p)
				}

Sub-question 2c

The worker-manager is a classic problem in parallel processing and can be used to model the task of CGI rendering of graphics in support of making movies. Please describe the worker-manager problem and explain the benefit of solutions to this problem for the task of CGI rendering.

Main question 3

Examination criterion: *Beskriva för- och nackdelar med olika tekniker för att lösa synkroniseringsproblem, inklusive semaforer, monitorer och tekniker för meddelandeöverföring*

Sub-question 3a

Describe the advantages and disadvantages of stateful vs stateless communication in the client-server architecture. Please give examples to illustrate your answer.

Sub-question 3b

RPC and Rendezvous are two communication mechanisms that work very well in a client-server architecture. Describe and discuss the advantages and disadvantages of these two approaches in the context of an airline booking service.

Sub-question 3c

Filters are one architecture within distributed computing and will often use message passing to be implemented. Describe the filter pattern with an example usage and discuss the advantages and disadvantages of asynchronous and synchronous message passing for this pattern.

Main question 4

Examination criterion: *Använda grundläggande tekniker såsom semaforer och meddelandeöverföring för att lösa synkronisering och kommunikation i program med parallella processer*

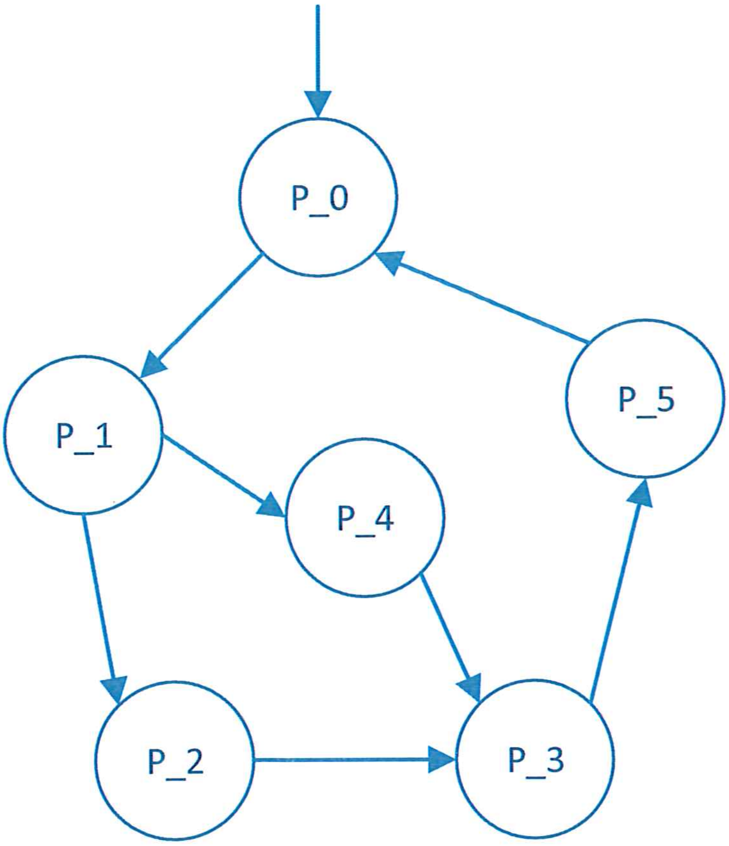
Sub-question 4a

A situation has arisen where there are three kinds of process A, B and C. Only 1 kind of process can be active at the same time, and up to 2 processes can be in their critical sections at the same time. Use semaphores and provide pseudo code this problem. You might not need to give full details for each kind of process but please explain why if you do not. Please remember to declare and initiate all semaphores and global variables that you need.

Sub-question 4b

What follows is a diagram of a set of processing nodes. Synchronization will be achieved using asynchronous message passing. Please state the number of channels required and give simple pseudocode for each process. The processes will continue to execute once started, as seen by the cycle of the connections, but should enter their critical sections only after **all** their preceding processes complete their execution.

P_0 is the first node that will execute, as seen by the initial arrow.



Sub-question 4c

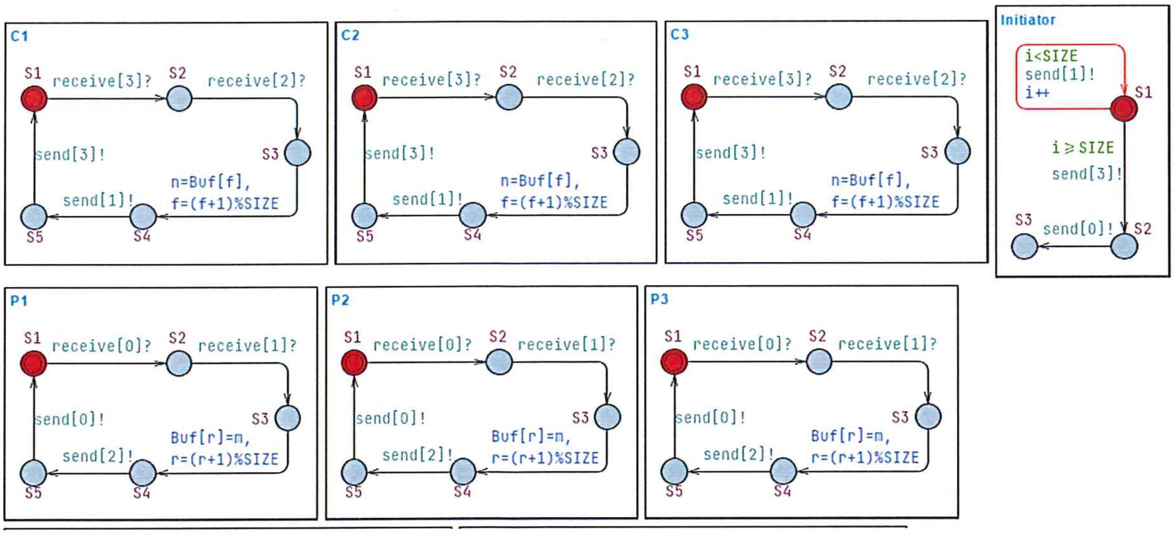
Implement a monitor in pseudo code for the dining philosopher's problem with 5 philosophers. The Philosophers look like this;

Process 1	Process 2

While True{
  <*your Monitor*>.start_eating(1)
  // do something
  <*your Monitor*>.stop_eating(1)
}
	While True{
	  <*your Monitor*>.start_eating(2)
	  // do something
	  <*your Monitor*>.stop_eating(2)
	}

Main question 5

Examination criterion: Modellera och verifiera egenskaper hos program med parallella processer, inklusive progression, frånvaro av låsning och ömsesidig uteslutning



Consider the above model for a producer / consumer problem with multiple processes of both types.

- ***Buf*** is a global array of size **SIZE** that function as a circular buffer.
- ***f*** and ***r*** are global variables that point out the current front and rear. The producer deposits in Buf[r] and the consumer fetches from Buf[f].
- ***n*** and ***m*** are local to the processes.
- The four channels are only used to synchronize the processes so no actual messages are sent through the channels (only empty messages).

Note that the system uses the emulated asynchronous messing passing mechanism that you used in assignment 2. This means that a sender does not block. You should assume that the processes emulating the four channels are part of the model even though they do not show in the figure.

Sub-question 5a

Explain the role of the initiator. I.e., what it does and why. Give an elaborate example from its start point at Initiator.S1 to its end point at Initiator.S3.

Sub-question 5b

Formulate an uppaal query to verify that two consumers cannot be in the critical section at the same time. Also, specify what your query is expected to return, true or false.

Sub-question 5c

Formulate an uppaal query to verify that a consumer and a producer can only be in the critical section at the same time if they access different elements in the buffer. Also, specify what your query is expected to return, true or false.