UNIVERSITY
OF SKÖVDE

School of Informatics

WRITTEN EXAMINATION

Course Concurrent Programming G1F, 7.5hp

Sub-course

Course code IT404G

Credits for written examination 5hp

Date 2026-03-28

Examination time 09:15-14:30

Examination responsible: Andras Marki

Teachers concerned

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.

Examination results should be made public within 18 working days

Good luck!

Total number of pages: 6

Parallella processer G1F, IT404G. 2026-01-08

Grading

The five main questions on the written exam correspond to the course objectives. Each main question comprises 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 fulfil 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{
	x=counter+2
	P(s)
	counter=x
	Print(x)
	V(s)
	}
		Int y=0;
		While true{
		P(s)
		counter--
		counter*=2
		y=counter
		V(s)
		}

Sub-question 1b

Define the two properties, fairness and avoiding unnecessary delays, in your own words. Ensure that your answer is sufficiently detailed to distinguish between the two properties.

Sub-question 1c

Semaphores can be implemented using busy wait mechanisms. What are the benefits and drawbacks of using a busy wait for semaphores?

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 the different processes, what each process does, and the synchronisation and communication mechanisms involved. For the mechanism you choose, please explain how it will work and **why** you chose it.

At the local pizzeria, people can order their pizza from the person at the desk. After the order is placed, the cook creates and bakes the pizza, then gives it to the person at the desk, who, in turn, serves the customer.

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 issues you observe. Message passing is asynchronous.

```
semaphore [] mxs=[ semaphore (1), semaphore (0), semaphore (1) ]
var a=None
```

```
processA{
  loop{
    var n<-f()
    P(mxs[2])
    P(mxs[0])
    a=n
    V(mxs[1])
    V(mxs[2])
  }
}
```

```
processB{
  loop{
    var n<-g()
    P(mxs[2])
    P(mxs[0])
    a=n
    V(mxs[1])
    V(mxs[2])
  }
}
```

```
processC{
  loop{
    P(mxs[1])
    var k=a
    V(mxs[0])
    h(k)
  }
}
```

Sub-question 2c

Describe the rules of the critical section problem. Describe what can happen if the rules are not obeyed, and give an example of a real-world problem where critical sections are necessary.

Parallella processer G1F, IT404G. 2026-01-08

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

Monitors provide implicit mutual exclusion when processes access them. Semaphores can also provide mutual exclusion. Describe these two and discuss the advantages and disadvantages of each.

Sub-question 3b

RPC and Rendezvous are two communication mechanisms that are well-suited to client-server architectures. Describe and discuss the advantages and disadvantages of these two approaches in the context of an airline booking service.

Sub-question 3c

Message passing can be either asynchronous or synchronous. Describe each of these. Compare and contrast the two mechanisms with respect to support for *mutual exclusion* and *efficiency*. Describe with examples where you could use each.

Parallella processer G1F, IT404G. 2026-01-08

Main question 4

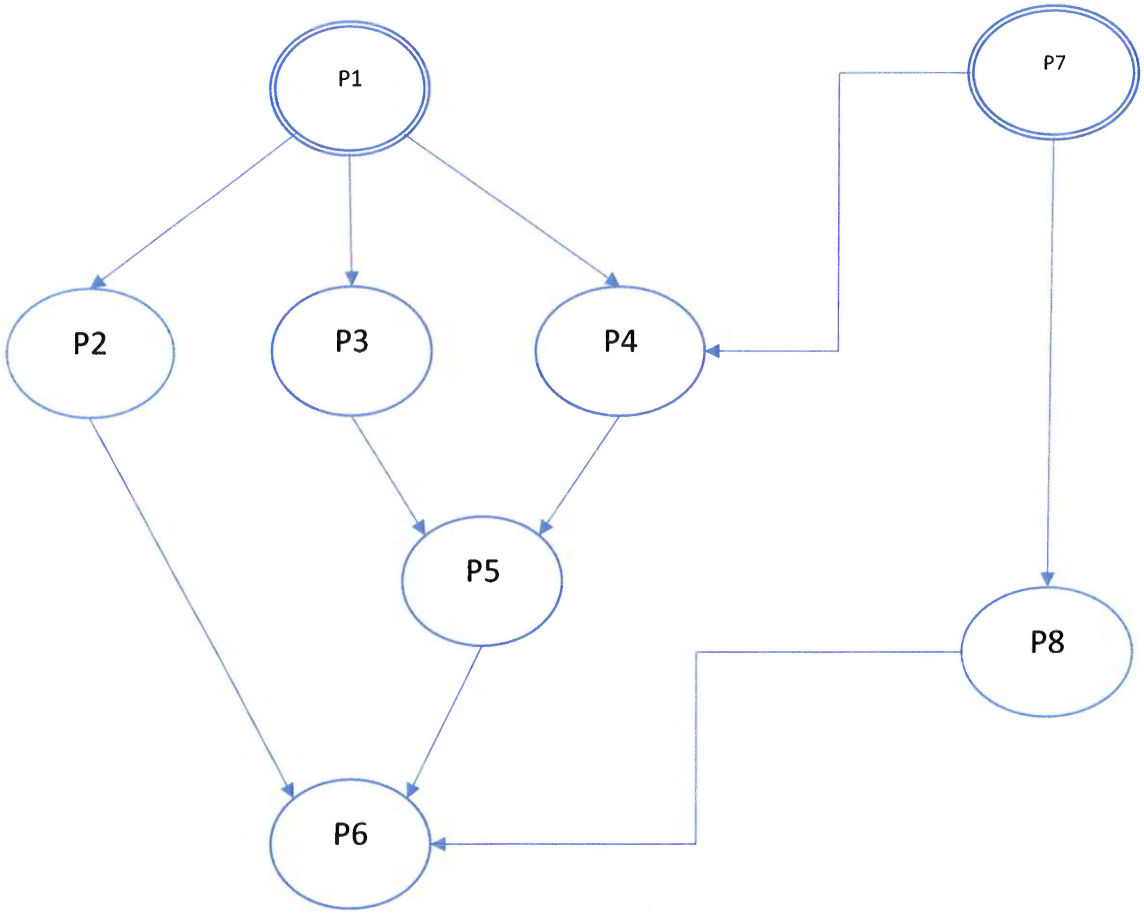
Examination criterion: *Använda grundläggande tekniker såsom semaforer och meddelandeöverföring för att lösa synkronisering och kommunikation i program med parallella processer*

Sub-question 4a

A situation has arisen in which there are three kinds of processes, A, B, and C. Only 1 process can be active at a time, and up to 2 processes can be in their critical sections simultaneously. Use semaphores and provide pseudocode as a solution to this problem. You might not need to provide full details for each 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. Synchronisation will be achieved using semaphores. Please state the number of semaphores required, their initial states and write pseudocode for each process. Please note that the arrows indicate that the later process should not begin until all earlier connected processes have completed. Nodes that can be assumed to be started positions are double ringed.



Sub-question 4c

The following is the code for 2 processes that implement the party-host problem. Implement the monitor that they both call.

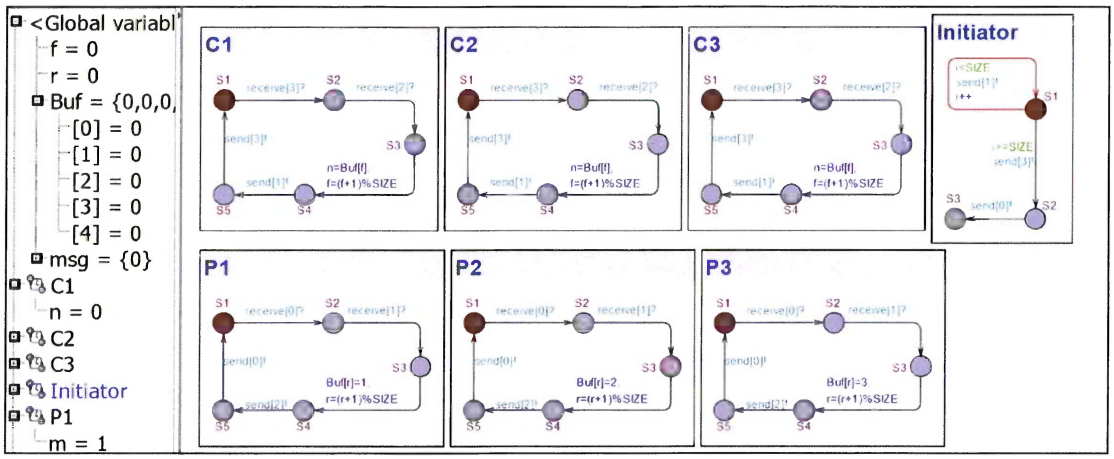
Monitor bowl	

process guest{
  loop{
  bowl.stand_in_front_of_bowl()
  if bowl.is_empty(){
  bowl.request_host()
  }
  else{
  bowl.take_a_drink()
  }
  bowl.walk_away_from_bowl()
  sleep(random)
  }
}
	Process host{
	  loop{
	  if bowl.requested_filling(){
	  bowl.stand_in_front_of_bowl()
	  bowl.fill_bowl()
	  bowl.walk_away_from_bowl()
	  }
	  sleep(random)
	  }
	}

Parallella processer G1F, IT404G. 2026-01-08

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 3 consumers and 3 producers. **Note** that the system uses an emulated asynchronous message-passing mechanism. There are 4 channel processes, which are not shown in this figure, but you should assume that they too are part of the complete system model and that they handle all send and receive operations just as in the assignments.

Explanation of diagram **Variables: Buf[SIZE] is global but n and m are local to the processes.** **Nodes: S1, S2, S3, S4 and S5 are names used for the different nodes. Start nodes are named S1.** **Edges: A transition from a node Si to Sj is described as Si to Sj**		
Processes C1, C2 and C3	**Processes P1, P2 and P3**	**Initiator**
S1 to S2: receive[3]? S2 to S3: receive[2]? S3 to S4: n=Buf[f]; f=(f+1)%SIZE S4 to S5: send[1]! S5 to S1: send[3]!	S1 to S2: receive[0]? S2 to S3: receive[1]? S3 to S4: Buf[r]=m; r=(r+1)%SIZE S4 to S5: send[2]! S5 to S1: send[0]!	S1 to S1: IF i<SIZE send[1]; i++ S1 to S2: IF i>=SIZE send[3] S2 to S3: send[0]

Sub-question 5a

Explain the purpose and the behaviour of the initiator.

Sub-question 5b

Formulate a Uppaal query to verify that C1 and C2 cannot be at node S3 simultaneously.

Sub-question 5c

Formulate a Uppaal query to verify that C1 and P1 can be in S3 only if they access different array entries.