UNIVERSITY
OF SKÖVDE

Cover sheet with information to the invigilator

School of: informationsteknologi

Course: Concurrent Programming

Examination

Course code: IT404G

Credits for written examination: 4.5

Date: 2024-01-10

Examination time: 5 hours

Available teacher: Richard Senington

Available on phone number: +46-702-67-8366

14.15 - 1930

Visiting the examination ☐ Yes, at
☒ No

Aids and other information for invigilators

Calculator ☐ Provided by the University
☐ Student's own calculator
☐ Not allowed

If you copy the exam papers yourself, provide the number of copies

Instructions to examinations responsible

All examination documents are to be handed in at Reprocentralen.

- **For copying of examination papers** the originals must be handed in no later than 6 workdays before the examination. The number of copies is filled in by Reprocentralen in the field below.
- **Copied examination papers** must be handed in no later than 3 workdays before the examination. Please notify the examination administration in due time when the papers will be handed in. Examination papers are to be handed over directly to the staff at Reprocentralen (not through mail). If you copy the exam papers yourself, provide the number of copies in the field above.

Hand-ins must be made during the opening times of Reprocentralen.

FILLED IN BY THE ADMINISTRATION

Number of copies 37

Number of sign-ups _____

UNIVERSITY
OF SKÖVDE

School of

WRITTEN EXAMINATION

Course: Concurrent Programming

Examination

Course code: IT404G

Credits for written examination: 4.5

Date: 2024-01-10

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

Parallella processer G1F, IT404G. 2024-01-10

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

One key property of concurrent systems is "absence of unnecessary delay. Describe unnecessary delays and what the consequences might be of not addressing it effectively, give examples to support your answer.

Sub-question 1b

Consider the following code for a simple lock between 2 threads. Will this work? If not, why not? Don't forget to motivate your answer with respect to all three safety properties, mutual exclusion, absence of deadlock and absence of unnecessary delay.

bool *flag1* = false bool *flag2* = false	

void lock(bool *myflag*, bool *itsflag*)
  myflag = true;
  while (*itsflag){*
  // skip
  }
}
	void unlock(bool myflag){
	  myflag=false;
	}

Sub-question 1c

Many synchronization mechanisms make use of queues to manage waiting threads. FIFO queues are the most common but implementations could use another kind of queue, such as a priority queue (where each thread indicates how important it is). Discuss the advantages and risks of this idea.

Parallella processer G1F, IT404G. 2024-01-10

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

In the below pseudo code please answer the following questions; what is the problem, is mutual exclusion ensured, is deadlock possible?

```
sem a,b,c = 1;
int x,y=0;
```

```
Process 1{
 Loop{
    P(a)
    P(b)
   Update(False)
    V(a)
    V(b)
    Sleep(random)
  }
}
```

```
Process 2{
 Loop{
    P(b)
    P(c)
   Update(False)
    V(c)
    V(b)
   Sleep(random)
  }
}
```

```
Process 3{
 Loop{
    P(c)
    P(a)
    Update(True)
    V(c)
    V(c)
   Sleep(random)
  }
}
```

```
Update(z){
 If(z){
   y-=x;
 }
 Else{
  y+=x+1;
 }
}
```

Sub-question 2b

int[] cs= [1,1,1,1,1,1,1]
channel c1,c2,c3

Type msg{String signal,int value}*

```
Process 1
loop{
 int x=nextCs()
 msg y
 if x==-1{
   send c2(-1)
   send c3 (-1)
   return
 }
 else{
  receive c1(y)
  if y.signal==”e”{
   cs[y.value]=1
  }else{
   If y.value==2{
     send c2 (x)
   }else{
     send c3 (x)
   }
  }
 }
}
```

```
Process 2
int y
loop{
   send c1(msg{“r”,2})
   receive c2 (y)
   if y==-1{
      return
   }
   // activity on y
   If error{
     send c1(msg{“e”,y})
   }
}
```

```
Process 3
int y
Loop{
   c1.send([“r”,3])
   receive c3(y)
   if y==-1{
      return
   }
   // activity on y
   If error{
     send c1(msg{“e”,y})
   }
}
```

```
Procedure nextCs
for int i=0;i<length(cs);i++{
  if cs[i]==1{
    cs[i]=0
    return i
  }
}
return -1
```

What is the pattern seen in this example code? What is the purpose of process 2 and 3 sending [“e”,y] and how does process 1 respond, in the context of the pattern you have identified.

**This is like a structure in C or a class in Java, a data type with 2 parts, signal and value*

Sub-question 2c

Describe the similarities and differences between the single buffer producer-consumer problem and a producer-consumer with 20 slots in a ring buffer.

Parallella processer G1F, IT404G. 2024-01-10

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

Mutual exclusion can be either implicit or explicit in different methods. Explain what this means and give examples of techniques that have each.

Sub-question 3b

Compare and contrast Conditional Critical regions with Semaphores. Your comparison should focus on support for mutual exclusion, efficiency and how easy it is to use the mechanism (i.e., to avoid mistakes).

Sub-question 3c

The Readers and Writers problem is a classic problem in concurrent programming which consists of 2 different kinds of process the reader and the writer. These two will both access a shared resource. The reader only reads the resource and several can read at the same time. The writer will change the resource and hence only one can have access to the resource at any one time. Additionally, a writer cannot be accessing the resource of a reader is also accessing it.

Semaphores and monitors can both be used for this problem. Compare and contrast each of these techniques for the problem. Make sure your comparison addresses the selective mutual exclusion that a solution to the R/W problem must guarantee.

Main question 4

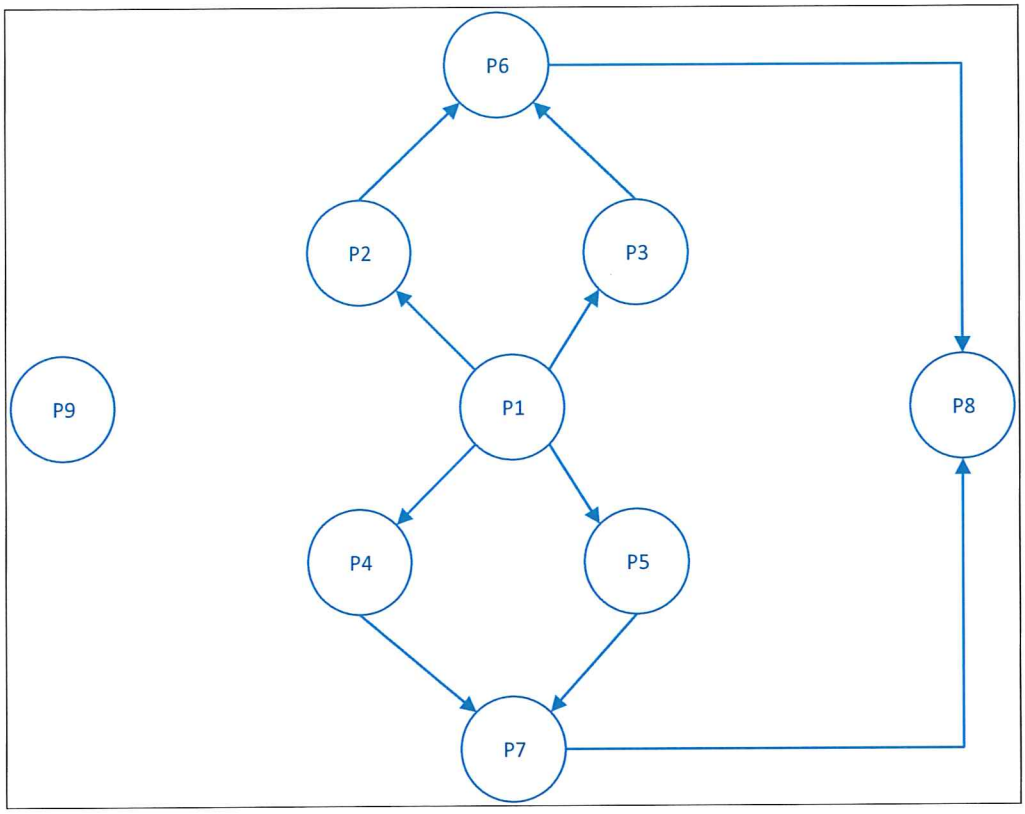
Examination criterion: *Använda grundläggande tekniker såsom semaforer och meddelandeöverföring för att lösa synkronisering och kommunikation i program med parallella processer*

Sub-question 4a

A home security company is designing the software for their alarm system. They have sensors on the doors and windows, an alarm and a control panel that the home owner can use to turn the system active or inactive. They want a system that will only activate the alarm if the system is currently active and they propose to use asynchronous message channels and the filter pattern to implement it. Design, graphically or in pseudo-code, this system with at least 1 sensor, the control panel and the alarm.

Sub-question 4b

What follows is a diagram of a set of processing nodes. Synchronization will be achieved using semaphores. Please state the number of semaphores required and write pseudocode for each process. Please note that the arrows mean that the later process should not begin until ALL the earlier connected processes have completed.



Sub-question 4c

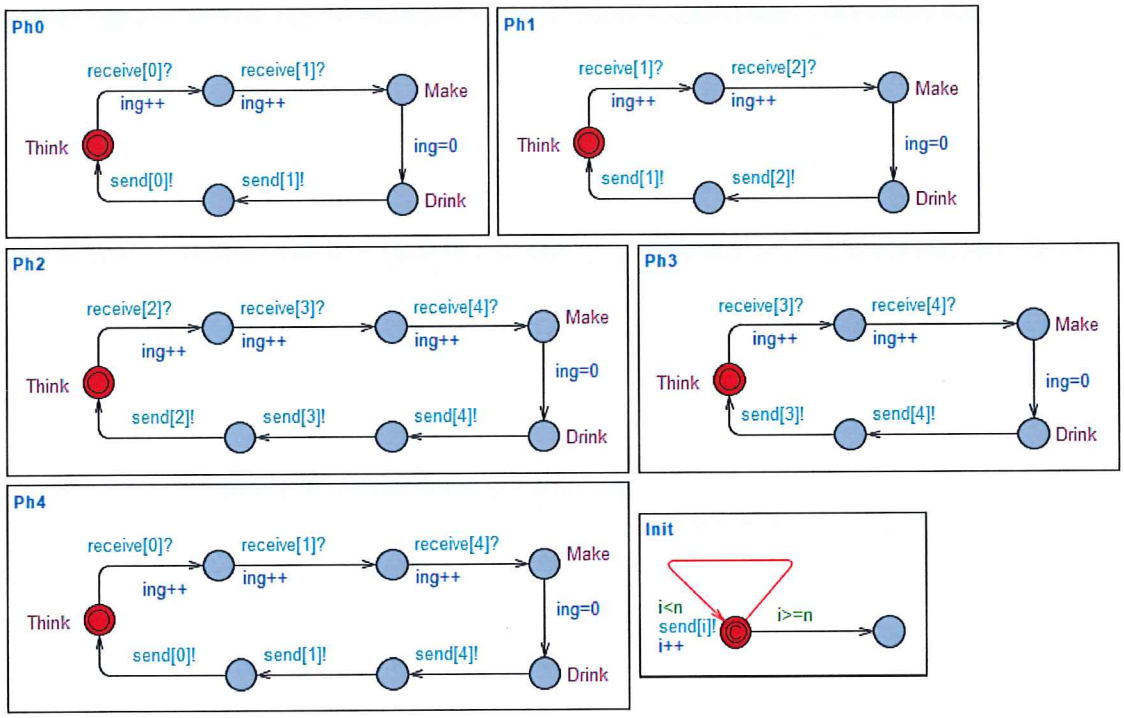
A bank wants to implement a “safe” account system. Each account will be managed by/protected by a Monitor. This Monitor will need 3 operations; withdraw, deposit, check_balance. They further propose that if someone tries to withdraw more than the current balance, that user/thread should be placed on a queue until the balance increases.

Design the Monitor.

Parallella processer G1F, IT404G. 2024-01-10

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 drinking philosophers. Each philosopher needs 2 or 3 bottles that they share with neighbors. **Note** that the system uses the emulated asynchronous message passing mechanism from assignment 2. There are 5 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. Here the channels are only used to guard the access to the bottles, i.e., to enforce mutual exclusion between neighbors. Therefore, no messages are sent.

Sub-question 5a

Explain the purpose and the behavior of Init. I.e., what it does and why.

Sub-question 5b

Formulate an uppaal query to verify that Philosopher Ph2 and Ph4 cannot be at node Make at the same time.

Sub-question 5c

Formulate an uppaal query to verify that Philosopher Ph4 can only be at node Make if he or she has exactly three ingredients.