Försättsblad till skriftlig tentamen vid Linköpings universitet

Datum för tentamen	2015-03-06
Sal (1)	TER2
Tid	8-12
Kurskod	TDDD82
Provkod	TEN2
Kursnamn/benämning **Provnamn/benämning**	Projekttermin inklusive kandidatprojekt: Säkra, mobila system Systemprogramvara: Skriftlig tentamen
Institution	IDA
Antal uppgifter som ingår i tentamen	4
Jour/Kursansvarig Ange vem som besöker salen	Simin Nadjm-Tehrani
Telefon under skrivtiden	0702 282412
Besöker salen ca klockan	9:15
Kursadministratör/kontaktperson (namn + tfnr + mailaddress)	Madeleine Häger
Tillåtna hjälpmedel	Inga
Övrigt	Rutigt papper
Antal exemplar i påsen	

Tentamen vid Institutionen för Datavetenskap, Linköpings universitet

TENTAMEN **TDDD82** Säkra mobila System (Systemprogramvara)

Datum: **2015-03-06**

Tid: 8 - 12

Lokal: TER2

Jourhavande-lärare:

- Simin Nadjm-Tehrani (endast via telefon **0702 282412**)

Hjälpmedel: Behövs inte.

Poängantal: 34p

Betyg: 3, 4 and 5:

U:	- 16 p
3:	17 - 22 p
4:	23 - 28 p
5:	29 - 34 p

INSTRUCTIONS:
Write your anonymous ID number on each sheet of paper that you hand in. Further, pages should only contain **answer to one question per page** (answers to sub-questions can be on the same page). You are asked to only answer on the front page of the paper. **Sort all the sheets** that you hand in, ordered in question number.

Your answers can be presented in Swedish or English.

Make sure your answers include motivations and are presented precisely. A correct answer **without any motivation will not be given any credits.** Incorrect answers embedded in a partially correct one reduce the points given for that answer. Points will not be given to answers that cannot be read due to bad handwriting.

Hints: Try to dispose of your time on each question in proportion of the assignment points. In those cases where you are in doubt about the question, write down your interpretation and your assumptions, and answer the question based on the interpretation.
Figures can be of help when describing but should be accompanied by a text description.

Results are reported no later than 24 March, 2015.

Good luck!

Simin Nadjm-Tehrani
Examiner TDDD82, Systems Software

Q1:

a) Semafor fungerar för att åstadkomma ömsesidig uteslutning endast om operationerna **wait** och **signal** implementeras som *atomiska operationer*. Beskriv två olika metoder för att få dessa operationer att bli atomiska och för varje metod ange en nackdel.

(4 poäng)

b) Consider the following pseudo code for the program consisting of the concurrent processes P1 and P2:

```
Process P1
   While true do {
     Read ()
     Rest ()
   }
```

```
Process P2
     While true do {
       Play ()
       write ()
     }
```

Give two example traces that belong to the program, and one example of a trace that does not belong to the program traces.

(3 poäng)

c) A system for monitoring forest fires consists of k sensors measuring different indicators in different parts of the forest. Each sensor runs parallel processes that read-and-send and receive acknowledgements from a central monitoring unit (a server) independently. Let N_1, ..., N_k be the sensor nodes that run in the north part of the forest and M be the monitoring node. The following pseudo code written by A. Newcomer shows the two processes running in each node N_i. The sensor readings are stored locally and sent in sequence followed by waiting for an acknowledgement (ack) for each sent value. Ack is a semaphore that is intended to be a condition variable. Assume that each sensor N_i starts by scheduling its process `Ni_producer_consumer` first. Assume further that the call of operation `receive` keeps the calling process at that program counter until it the receiving of a message is completed.

```
Sem: Ack (initially 0)
Int: count_i (initially 0)
Process Ni_producer_consumer
while true do {
    read(value);
    store(value, buffer);
    count_i = count_i + 1;
    send((count_i, value), M);
    wait(Ack)
  }
```

```
Process Ni_Ack_observer
while true do {
       receive((count_i, ack), M);
       signal(Ack);
}
```

If the server is slow due to the large number of values being sent from large numbers of sensors, what is the consequence for this implementation at a given sensor node N_i? If you find a shortcoming with the current structure of the code, how would your revised pseudo code look like?

(3 poäng)

d) Give one reason why using a monitor may be preferable to other methods for implementing mutual exclusion in a concurrent system. Give one reason why using a monitor might not be possible and one may have to resort to other methods for implementing mutual exclusion.

(4 poäng)

e) Consider the dining philosopher scenario for which one philosopher has a code different from the other four philosophers. This philosopher takes the right fork first and then the left fork (as two separate operations), whereas all other philosophers take the left fork first followed by the right fork. Doe this system have a trace that leads to deadlock? If so, present this trace. Otherwise, motivate your answer by referring to the Coffman conditions.

(4 poäng)

Q2:

a) Ta ställning till följande utsagor (sant eller falsk), och motivera ditt svar med hänvisning till definitioner av termer.
   1) It is enough for a network to provide high throughput in order to guarantee the right QoS for voice communication.
   2) Diffserv is preferred over Intserv since it guarantees service level agreements.

(4 poäng)

b) Consider an air traffic control system where the goal of distributed traffic controller nodes is to collectively keep flying aircrafts within safety distances from each other. Decide which one of the following properties is a functional property and which is an extra-functional property. Motivate your answer!

   1. The controller node should accept flight plans from each running aircraft and monitor the current position of the aircraft on a common screen.
   2. The amount of flight data regularly transmitted by each aircraft within a controller's coverage range should be limited to X kb.
   3. If a flight data for one aircraft is lost the controller node should be able to retrieve the information on a second (redundant) poll or through contacting the other aircrafts in vicinity.

(3 poäng)

Q3:

a) Present two concrete instances of fault models (according to classification crash, omission, timing, Byzantine, …) in the *distributed* scenario described in Q2 b) and provide a method that reduces the impact of each fault. For one of the methods you chose in this question relate the method to the notions of redundancy in space (software/hardware/data) or redundancy in time.

(3 poäng)

b) Provide two variations of client server architecture by describing the elements of the architecture and an example application (for each variation) that uses this architecture.

(2 poäng)

Q4:

Använd begreppen från IFIP Working Group 10.4 för att analysera felkälla-felyttring-haveri i detta sammanhang, och klassifiera felkällan som permanent/transient/intermittent.

a) The US Federal Communications Commission (FCC) has issued an investigative report on last April's 911 meltdown that affected 81 public safety answering points (PSAP) in seven states and blocked over 5,600 calls for help, saying the event was entirely preventable. It explained that the outage occurred because of a "software coding error" at Intrado Inc.'s Colorado network center, which provides routing services for several states.

"At 11:54 p.m. on April 9, 2014, the PSAP trunk member's counter at Intrado's emergency call management centre exceeded its threshold and could send no more 911 calls to PSAPs using CAMA[1] trunks. Under normal operations, the PSAP trunk member assigns a unique identifier for each call that terminates using CAMA trunks. This is how Intrado has implemented the protocol commonly used to complete 911 calls over CAMA trunks, which (unlike SS7) require additional features to carry the signaling.
In this case, the trunk assignment counter reached a pre-set capacity limit to assign trunks, which meant that no additional database entries to reserve a PSAP CAMA trunk could be created, no trunk assignments for call delivery could be made for PSAPs with CAMA trunks and, therefore, no 911 calls could be completed to these PSAPs or any backup PSAP through the Englewood emergency call management centre.
When the software stopped making trunk assignments, it prevented calls being routed through the Englewood hub from reaching these PSAPs. Further, inadequate alarm management resulted in significant delays in determining the software fault and restoring 911 service to full functionality. Intrado operated a redundant hub in Miami, Florida to which 911 traffic could have been immediately rerouted, but because the malfunction was not detected promptly and mitigation actions were not efficiently developed, Intrado did not execute either an automatic or manual switchover of traffic to the Miami hub until six hours had elapsed. This switchover almost immediately restored the service."

(4 poäng)

[1] CAMA: Centralised Automatic Message Accounting

Glossary: English to Swedish

atomic – atomisk (oavbrytbar)
availability – tillgänglighet
avoid – undvika
bandwidth – bandbredd
broker – medlare
clock drift – klockdrift
concurrency – samtidighet
consistency – konsistens
deadlock – låsning (baklås)
delay – fördröjning
deliberate – avsiktlig
dependability – pålitlighet
diversity – mångfald
error – felyttring
event – händelse
failure – haveri
fault – felkälla
fault tolerance – feltolerans
forecast – förutse
inheritance – arv
integrity – dataintegritet
interleavings – sammanflätningar
interoperability – interoperabilitet
intrusion – intrång
latency - latens
maintainability – reparerbarhet
malicious - illvillig
middleware - mellanvara
mutual exclusion – ömsesidig uteslutning
necessary condition – nödvändigt villkor
non-functional property – ickefunktionell egenskap
omission – utelämnande
performance – prestanda
preemptible – avbrytbar
prevent – förebygga
quality of service – tjänstekvalitet
race condition – kapplöpningstillstånd
release - släppning
reliability – tillförlitlighet
rendundancy – redundans
replica – kopia
replication – replikering
request – begäran
response time – responstid
safety – säkerhet
scheduler – schemaläggare
security – säkerhet
serialisable - serialiserbar
shaping - formning
shared memory – gemensamt minne
starvation – svält
sufficient condition- tillräckligt villkor
synchronisation – synkronisering
system call – systemanrop
thread - tråd
throughput – genomströmning
trace – spår
transparency – transparens
trust – tillit
validation – validering
verification - verifiering
vulnerability - sårbarhet

Swedish – English

arv - inheritance
atomisk (oavbrytbar) – atomic
avbrytbar - preemptible
avsiktlig - deliberate
bandbredd - bandwidth
begäran - request
dataintegritet - integrity
felkälla - fault
feltolerans - fault tolerance
felyttring - error
fördröjning - delay
förebygga - prevent
formning - shaping
förutse - forecast
gemensamt minne - shared memory
genomströmning - throughput
händelse - event
haveri - failure
ickefunktionell egenskap - non-functional property
illvillig - malicious
interoperabilitet - interoperability
intrång – intrusion
kapplöpningstillstånd – race condition
klockdrift - clock drift
konsistens - consistency
kopia - replica
låsning (baklås) - deadlock
latens - latency
mångfald - diversity
medlare - broker
mellanvara - middleware
nödvändigt villkor - necessary condition
ömsesidig uteslutning - mutual exclusion
pålitlighet - dependability
prestanda - performance
redundans - rendundancy
reparerbarhet - maintainability
replikering - replication
responstid - response time
säkerhet - safety
säkerhet - security
sammanflätningar - interleavings
samtidighet - concurrency
sårbarhet - vulnerability
schemaläggare - scheduler
serialiserbar - serialisable
släppning – release
spår - trace
svält - starvation
synkronisering - synchronisation
systemanrop - system call
tillförlitlighet - reliability
tillgänglighet - availability
tillit - trust
tillräckligt villkor - sufficient condition
tjänstekvalitet - quality of service
tråd - thread
transparens - transparency
undvika - avoid
utelämnande - omission
validering - validation
verifiering - verification