Tentamen vid Institutionen för Datavetenskap, Linköpings universitet

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

Date: **2016-03-04**

Time: 14 - 18

Room: TER1

On call:
- Mikael Asplund (telephone: **0700 895 827**)

Aids: Not needed.

Max number of points: 34p

Grades: 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.

Good luck!

Mikael Asplund
Examiner TDDD82, Systems Software

Q1 (17 points):

Consider the following pseudo code for a program that is composed by the processes P1 and P2. Assume that every statement in the code is atomic and that all possible interleavings are equally likely to occur.

```
Process P1 {
   X := 1
   Y := 2
   if (X == Y) {
      print("1")
   }
}
```

```
Process P2 {
   X := 2
   Y := 2
   if (Y == X) {
          print("2")
   }
}
```

a) How many possible interleavings can occur when running the program? (1 point)

b) What are the possible outputs of the program that can actually occur? Motivate each output why or why not it is a possible output of the program. (3 points)

c) What is the probability that the output of the program is "21"? (4 points)

d) Use semaphores to ensure that the output of the program is "12". (2 points)

e) Explain the use of condition variables in monitors with the help of an example. (2 points)

f) Briefly explain each of the Coffman conditions. (2 points)

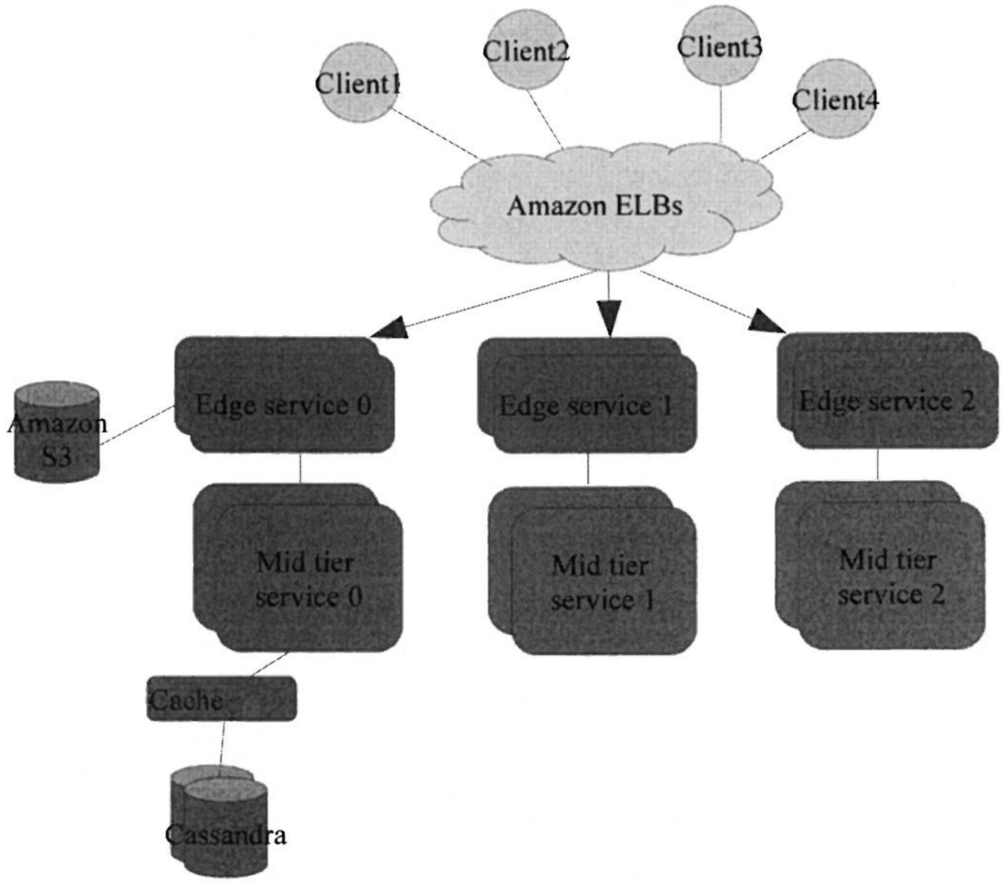
g) Consider the following resource allocation problem in a system with three resources (R1-R3), and four processes (P1-P4). The table indicates the currently allocated resources and in parenthesis the maximum possible demand (e.g., P1 currently has 2 instances of resource R1 and can at most request a total of 3 instances).

	Resource R1	Resource R2	Resource R3
P1	2 (3)	2 (3)	0 (1)
P2	1 (1)	0 (2)	0 (0)
P3	1 (1)	2 (2)	0 (1)
P4	0 (4)	1 (1)	0 (1)

In total there are 4 instances of resource R1, 7 instances of resource R2, and 1 instance of resource R3. Use Banker's algorithm to determine if the request from P2 of 2 instances of resource R2 should be granted. (3 points)

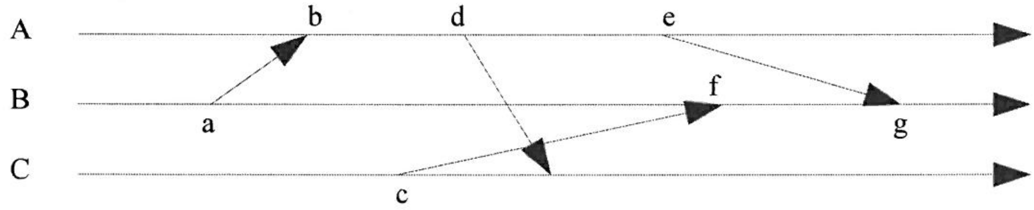
Q2 (8 points):

a) The figure below is a simplified reconstruction from a figure in the Netflix Tech blog http://techblog.netflix.com/2012/06/annoucing-archaius-dynamic-properties.html with the description that it "shows a **hypothetical simplistic overview** of a typical deployment architecture at Netflix". Explain how this architecture makes use of horizontal and vertical distribution (note that service 0 and 1 are really different services, not just copies of the same service). (3 points)



b) Explain why it might be appropriate to use an asynchronous system model when designing a distributed system. (2 points)

c) Use Lamport clocks to give timestamps to the events (a-g) in the figure below (assume each node starts with a clock value 0):



(3 points)

Q3 (5 points):

a) Jitter is an important QoS metric in network and real-time application. Provide two alternative ways do define jitter. (2 points)

b) Consider a system for monitoring the health of elderly patients in their homes. For each of the following system requirements take a stand on whether it is a functional or extra-functional requirement (remember to motivate your answer):
   - If the patient pushes the alarm button, then emergency services should be notified.
   - The expected frequency of an alarm not reaching Emergency services due to network problems should be less than once every 10 years.
   - Personal information of patients should only be accessible by the authorised medical personnel.

(3 points)

Q4 (4 points):

Use the terminology from IFIP Working Group 10.4 to analyse the fault-error-failure chain in the example below. Classify the fault as permanent/transient/intermittent".

"While driving in autonomous mode, a Google self-driving car was involved in a minor accident with a public transit bus in California on Valentine's Day, according to an accident report (PDF) filed with the California Department of Motor Vehicles (DMV):

A Google Lexus-model autonomous vehicle ("Google AV") was traveling in autonomous mode eastbound on El Camino Real in Mountain View in the far right-hand lane approaching the Castro St. intersection. As the Google AV approached the intersection, it signaled its intent to make a right turn on red onto Castro St. The Google AV then moved to the right-hand side of the lane to pass traffic in the same lane that was stopped at the intersection and proceeding straight.

However, the Google AV had to come to a stop and go around sandbags positioned around a storm drain that were blocking its path. When the light turned green, traffic in the lane continued past the Google AV. After a few cars had passed, the Google AV began to proceed back into the center of the lane and pass the sandbags. A public transit bus was approaching from behind.

The Google AV test driver saw the bus approaching the left side mirror but believed the bus would stop or slow to allow the Google AV to continue. Approximately three seconds later, as the Google AV was reentering the center of the lane it made contact with the side of the bus. The Google AV was operating in autonomous mode and traveling less than 2 mph, and the bus was traveling at about 15 mph at the time of contact.

The Google AV sustained body damage to the left front fender, the left front wheel and one of its driver's-side sensors. There no injuries reported at the scene."

(4 points)

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