

# EDA095

## Tentamensskrivning — Nätverksprogrammering 2005-05-27, kl. 8-13

### DEL 1 – Frågor av teoretisk, principiell eller utredande karaktär

#### Anvisningar

Tillåtna hjälpmedel för denna del av tentamen: **Inga**. Kurslitteratur och andra hjälpmedel för del 2 av tentamen skall förvaras på golvet bredvid bordet eller vid salens vägg.

Denna tentamen i kursen Nätverksprogrammering består av två delar: en del som innehåller frågor av teoretisk/principiell/utredande karaktär och en del som innehåller praktiska programmeringsuppgifter. Detta är del 1. När du löst uppgifterna i denna del av tentamen lämnar du in din lösning i det vita tentamensomslaget varvid du erhåller del 2 av tentamen tillsammans med ett nytt, färgat, tentamensomslag som skall användas vid inlämning av din lösning på del 2 av tentamen.

För godkänt betyg på tentamen krävs sammanlagt minst 20 poäng på tentamen, varav minst 8 poäng på vardera deltentamen. För högre betyg krävs naturligtvis mer, så gör så många uppgifter du kan.

Senast 2005-06-01 anslås på institutionens anslagstavla vilka som deltagit i tentamen men som, enligt institutionens noteringar, ännu inte redovisat laborationerna eller projektet. Deras skrivning rättas inte (och kan på sikt komma att annulleras) såvida inte rättelse skett eller dispens erhållits från ansvarig lärare. Rättelse skall göras senast 2005-06-15.

## Uppgifter

### 1.

- a) Vad står följande två förkortningar för:
- URL
  - XML
- (1 p)
- b) Jämför ett serverprogram som använder TCP (eller UDP) med ett som använder RMI. Hur skiljer de sig åt? (2 p)

2. Vad är en servlet? Beskriv kortfattat hur de fungerar. (2 p)

3. Beskriv vilka steg man följer för att skriva ett enkelt server- och ett enkelt klientprogram med RMI. Med enkelt program menar vi utan kodöverföring. Beskriv också vilka steg som krävs för att sätta upp servern. (2 p)

### 4. XML

- a) Man beskriver strukturen av ett XML-dokument i en DTD där man specificerar relationer mellan tre huvudkomponenter. Vad är namnet på dessa tre komponenter? (1 p)
- b) Vad betyder dessa två begrepp i XML-validering:
- *well formed*
  - *valid*.
- (1 p)
- c) I ett XML-dokument använder man ett kodningsschema (*encoding*). Namnge ett sådant kodningsschema som kan användas i XML. (0,5 p)
- d) Man kan använda Document Object Model (DOM) för att arbeta med XML-dokument. Vad får man för sorts (data-)struktur som resultat från en DOM-parser? (0,5 p)
- e) Nämn en operation som man kan applicera på strukturen i förra deluppgiften. Ge också ett exempel på en applikation där operationen du nämnt kan användas. (1 p)

### 5. RTP

The Real Time Protocol (RTP) har fyra viktiga fält: *payload*, *sequence number*, *timestamp* och *SSRC*. Vilka av följande fyra påståenden är korrekta?

- a) **Payload** beskriver hur en service provider fakturerar kunder. (0,5 p)
- b) **Sequence number** är en räknare som ökar med ett varje gång sändaren skickar ett paket. (0,5 p)
- c) **Timestamp** nollställer klockan i början och ökar med ett för varje paket. (Det är faktiskt en duplicering av *sequence number* för kontrollsyfte) (0,5 p)
- d) **SSRC** identifierar strömningssessionen. (0,5 p)

### 6. Javatrådar

- a) Förklara Javas monitorbegrepp. Vad är syftet med att det finns och vilka element i språket utgörs det av. (1 p)
- b) I vilka situationer har man nytta av att kunna ge trådar olika prioritet? (0,5 p)

7. Asynkron kommunikation mellan trådar kan ske via en så kallad *mailbox*. En mailbox består i regel av två metoder; `put()` och `get()`.

En programmerare har ett program med flera så kallade *producent*-trådar, som genererar textmeddelanden som ska lösas av en så kallad *konsument*-tråd och vill att trådkommunikationen ska ske via en mailbox. Programmeraren, som inte läst nätverksprogrammering, har gjort följande mailbox-implementation:

```
public class Mailbox {
    private String[] box;
    private int msgNbr;

    public Mailbox(int size) {
        box = new String[size];
        msgNbr = 0;
    }

    public void put(String msg) {
        while (msgNbr == box.length) {
            // Wait until the mailbox isn't full
        }
        box[msgNbr++] = msg;
    }

    public String get() {
        while (msgNbr == 0) {
            // Wait until the mailbox isn't empty
        }
        return box[--msgNbr];
    }
}
```

- Vad brukar tekniken, som används i `put()` ovan, kallas och vad finns det för nackdelar med tekniken? (1 p)
- Det kan uppstå så kallad *kapplöpning* (race condition) i ovanstående program. Vad innebär det och hur kan det åtgärdas? (1 p)
- Om det finns många producenttrådar och en konsumenttråd, som dessutom har lägre prioritet än producenttrådarna, kan det hända att konsumenttråden inte får någon CPU-tid. Vad kallas detta fenomen? (0,5 p)
- Skriv om mailbox-implementationen ovan så att problemen nämnda i deluppgift a och b försvinner. Trådarna, som anropar klassen, ska inte behöva skrivas om. (3 p)

