LUNDS TEKNISKA HÖGSKOLA Institutionen för datavetenskap

Tentamen
Nätverksprogrammering
Del 1

2007–05–22, 8.00–13.00

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 mer, så gör så många uppgifter du kan.

---

1. När man använder strömmar i form av en TCP-uppkoppling är det ofta fördelaktigt med att använda en *buffrad* ström.

   a) Nämn en fördel med buffring av strömmar.

   (1p)

   b) Buffring kan också medföra problem om man inte ser upp. Beskriv ett scenario där buffring kan medföra att kommunikationen mellan två datorer via TCP inte fungerar. Hur löser man problemet i Java?

   (2p)

2. I samband med XML pratar man ibland om förkortningen DTD.

   a) Vad står förkortningen DTD för?

   (1p)

   b) Vilken typ av information hittar man i en DTD?

   (1p)

3. Vilka av följande påståenden är sanna?

   a) RTP är ett protokoll som garanterar leverans av ljud-/bilddata i realtid.

   b) SIP används för att starta och konfigurera en nätverksförbindelse.

   c) RTCP står för Real-time Transmission Control Protocol.

   d) En *codec* är en komponent som krypterar känslig information innan den överförs över nätverket så att den inte kan avlyssnas.

   e) SDP är ett protokoll som används tillsammans med SIP för att specificera olika parametrar för en nätverksförbindelse.

   f) RTP förlitar sig på att det finns ett applikationslager som kompenserar för jitter och paketförluster.

   (3p)

---

4. Redogör i stora drag för hur RMI fungerar, med avseende på vilka nätverksuppkopplingar och vilka data som skickas, från det att en klient startas och det att det första RMI-anropet slutförts. Vilka programvarukomponenter är inblandade? Hur överförs parametrar/funktionsresultat?

(3p)

5. Nämn två olika välkända webbtekniker för att generera dynamiska HTML-sidor på serversidan.

(1p)

6. Vilken uppgift har en så kallad *protocol handler* i Javas klassbibliotek för nätverkskommunikation?

(1p)

7. Vad är *multicast* och vilka förutsättningar (nämn två) måste vara uppfyllda för att multicasttrafik ska nå fram till den avsedda mottagaren?

(2p)

8. I ett javaprogram behöver en tråd vänta (en gång) tills en annan parallell tråd har avslutat någon typ av beräkning som den första tråden är beroende av innan den kan fortsätta. I ett första förslag till lösning har man skrivit följande Javakod för tråden (för problemet irrelevant kod har ersatts med "..."):

```
public class WaitingThread extends Thread {

    private boolean finished = false;

    public void releaseThread() {
        finished = true;
    }

    public void run() {
        ...

        /* Wait for computation to finish. */
        while (!finished) {}

        /* The external thread has now finished its computation */
        ...
    }
}
```

Koden fungerar så att den parallella, externa, tråden anropar metoden `releaseThread()` när den aktuella beräkningen har avslutats. Om tråden `WaitingThread` kommer fram till while-satsen i metoden `run()` innan den parallella tråden har exekverat `releaseThread()` kommer while-satsen att göra att trådens exekvering hejdas (dvs att efterföljande rader inte exekveras). Omvänt, om `releaseThread()` anropats innan while-satsen nåtts kommer exekveringen *inte* fördröjas.

a) Vad kallas det sätt som används ovan för en tråd att vänta på en annan tråd?

(1p)

b) Metoden för väntan i deluppgift a) har en stor nackdel, vilken?

(1p)

c) Skriv om klassen `WaitingThread` ovan med hjälp av Javas inbyggda monitorbegrepp så att problemet i deluppgift b) elimineras.

(3p)