

Tentamen

Nätverksprogrammering

Del 1

2009-06-01, 14.00-19.00

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. Ange för varje påstående nedan om det är sant eller falskt.

- a) Om vi vill skapa en TCP-förbindelse till en server på en annan dator måste vår lokala socket vara bunden till samma portnummer på vår lokala maskin som servern använder på den andra datorn.
- b) Buffring är en teknik som kan dramatiskt förbättra genomströmningen av data på en ström men kan också i särskilda fall ge upphov till dödlägen i kommunikationen.
- c) När en JSP-sida begärs från en webbserver kommer servern att automatiskt konvertera JSP-sidan till en servlet, om nödvändigt, och sedan exekvera denna.
- d) Förkortningen RFC står för ett protokoll som används för strömmande ljud eller video.
- e) Värdet kallat *Time-To-Live* är i multicastsammanhang ett mått i sekunder på hur länge informationen i ett paket ska betraktas som aktuellt.
- f) PHP kan användas som ett generellt skriptspråk och inte bara för att generera webbsidor.
- g) Om en tråd exekverar en metod `a()`, som är deklarerad `synchronized`, i ett objekt kan en annan tråd samtidigt exekvera samma metod `a()` i ett annat objekt av samma klass.
- h) Om en tråd exekverar en metod `a()`, som är deklarerad `synchronized`, i ett objekt och koden för metoden `a()` i sin tur innehåller ett anrop till en annan metod i samma objekt som även den är deklarerad `synchronized`, kommer tråden att hamna i ett dödläge eftersom objektet inte kan låsas två gånger.

(4p)

2. Vi har pratat om något som kallas för URL:er i kursen.

- a) Vad är en URL (*inte* javaklassen URL) och vilka är dess inre beståndsdelar?

(1p)

- b) Vad är huvudskillnaden mellan javaklasserna URL och URI?

(1p)

3. I samband med RMI har vi pratat om ett begrepp kallat *marshalling*. Vad menar vi med det?

(1p)

4. I samband med att Java 1.4 kom infördes ett nytt paket i Javas standardbibliotek kallat `java.nio`. Detta nya paket ger bland annat stöd för att på ett effektivt sätt hantera flera oberoende uppkopplingar parallellt, i en server till exempel, utan att man behöver tillgripa multitrådning. Detta är funktionalitet som länge har funnits på operativsystemnivå, men som det inte funnits stöd för i Java.

En klass och en metod som har en nyckelroll i sammanhanget är klassen `Selector` och metoden `select()`. Redogör kortfattat, och i övergripande ordalag, för hur klassen och metoden används för att man ska kunna hantera flera oberoende parallella uppkopplingar utan att tillgripa multitrådning. Se till att ditt svar berättar vad metoden gör rent tekniskt samt hur detta hjälper oss i den aktuella uppgiften.

(2p)

5. För att parsea HTML- eller XML-dokument finns det två huvudtekniker i Java. Den ena tekniken används av `HTMLEditor.Parser` för att parsea HTML och den andra tekniken kallas *DOM*. Beskriv översiktligt de två olika parserteknikerna och hur de skiljer sig åt för programmeraren som använder dem.

(3p)

6. RTP (Real Time Transport Protocol) och liknande protokoll har utvecklats för att komma till rätta med några av de problem som uppstår om man försöker använda TCP/UDP för överföring av strömmande ljud/bild över Internet. Nämn två problem med TCP/UDP som man måste ta hänsyn till vid överföring av strömmande ljud/bild.

(1p)

7. När man skriver ett program i Java som består av flera samverkande trådar uppstår ofta ett behov av att synkronisera trådarnas exekvering. Till exempel kan en tråd behöva vänta på att en annan tråd ska bli färdig med någon beräkning eller så behöver man se till att inte två trådar försöker använda sig av samma resurs samtidigt. En ofta använd teknik i Java är att använda sig av så kallade monitorer. För att stödja användandet av monitorer har man i Java infört ett antal reserverade ord och standardmetoder.

a) Beskriv vad följande reserverade ord/standardmetoder har för innebörd (semantik) i Java:

- 1) `wait()`
- 2) `notify()`
- 3) `notifyAll()`

(2p)

b) För att man ska kunna anropa metoderna `wait()`, `notify()` eller `notifyAll()` (utan att få ett exception). Måste ett mycket specifikt villkor vara uppfyllt. Vilket?

(1p)

8. En programmera vill skriva ett enkelt program i Java som lyssnar efter multicastmeddelanden som sänds till multicastadressen "experiment.mcast.net" på port 4099. Varje meddelande innehåller en text (i Javas standardkodning) som ska skrivas ut. Alla texter är tillräckligt korta för att få plats i ett UDP-paket. Hans kod visas nedan:

```
import java.net.*;
import java.io.*;

public class MCReader {
    public static void main(String args[]) {
        try {
            MulticastSocket ms = new MulticastSocket(4099);
            InetAddress ia = InetAddress.getByName("experiment.mcast.net");
            byte[] buf = new byte[65536];
            DatagramPacket dp = new DatagramPacket(ia, 4099, buf, buf.length);
            while(true) {
                ms.receive(dp);
                String s = new String(dp.getData(), 0, dp.getLength());
                System.out.println("Received: " + s);
            }
        } catch(IOException e) {
            System.out.println("Exception: " + e);
        }
    }
}
```

- a) När programmet testkörs visar det sig att inga meddelanden kommer fram över huvud taget. Varför är det så och hur borde programmet ändras så att meddelandena kommer fram? Det räcker om du beskriver ändringen med vanliga ord eller med pseudokod om du inte kommer ihåg exakt hur det skulle se ut i Java.

(2p)

- b) När ändringen ovan är genomförd och meddelanden äntligen kommer fram upptäcks det att meddelandena som skrivs ut är ofullständiga (trunkerade). Varför är det så och hur borde programmet ändras så att meddelandena kommer fram? Det räcker om du beskriver ändringen med vanliga ord eller med pseudokod om du inte kommer ihåg exakt hur det skulle se ut i Java.

(2p)

Slut på del 1 – lämna in och hämta ut del 2!