

Tentamen

Nätverksprogrammering

Del 1

2008-05-26, 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. Torrent-Tore har skrivit en fildelningsserver som förmedlar data från en fjärransluten filserver till en klient ansluten till fildelningsservern. Som en del av koden är det nödvändigt att läsa in bytes från en öppen TCP-förbindelse och vidareförmedla dem oförvanskade till en annan TCP-förbindelse tills inget mer finns att läsa eller ett fel uppstår. De båda förbindelserna representeras av streamobjekten `input` respektive `output`. Utdrag ur Torrent-Tores kod:

```
...
try {
    int sent = 0;
    byte[] buffer = new byte[128];
    while (sent >= 0) {
        sent = input.read(buffer);
        if (sent >= 0) {
            output.write(buffer);
            output.flush();
        }
    }
} catch (Exception e) {
    ...
}
...
```

- a) Koden som Torrent-Tore skrivit fungerar dock inte riktigt som avsett. Varför?

(1p)

- b) Hur borde Torrent-Tore modifiera sin kod för att åtgärda problemet?

(1p)

2. Teknikerna *servlets* och *JSP* är intimt knutna till varandra. På vilket sätt?

(1p)

-
3. I HTTP, protokollet för kommunikation med en webbserver, förekommer begreppet *frågesträng* (eng. *query string*) vilket har betydelse när man skapar dynamiska webbsidor på webbservern.
- a) Ge ett exempel på information som kan överföras via en frågesträng (eng. query string).
(1p)
- b) I HTTP hanteras frågesträngen olika beroende på om ett GET-kommando eller ett POST-kommando används. Redogör kortfattat för skillnaden mellan de två fallen på HTTP-nivå.
(1p)
4. För att kommunicera över TCP i Java använder man traditionellt klasserna `Socket`, `ServerSocket`, `InputStream` och `OutputStream`. I Java 1.4 introducerades dock paketet `java.nio` med bl.a. klasserna `SocketChannel`, `ServerSocketChannel`, `Selector`, med flera.
- a) Nämn en betydande fördel som kan finnas med att använda det nya paketet (`java.nio`) i stället för det gamla.
(1p)
- b) Vilken roll har klassen `Selector` i paketet?
(1p)
5. Datagram-David har skrivit följande kodsutt i Java för att ta emot UDP-paket med varierande längd från `DatagramSocket`-objektet `socket` och skriva ut innehållet (tolkat som en sträng). Kodsnutten innehåller dock ett signifikant logiskt fel.

```
...
try {
    byte[] buf = new byte[65507];
    DatagramPacket dp = new DatagramPacket(buf, buf.length);
    while (true) {
        socket.receive(dp);
        System.out.println(new String(dp.getData(), 0, dp.getLength()));
    }
} catch (Exception e) {
    System.out.println(e);
    System.exit(1);
}
...
```

- a) Hur kommer felet att yttra sig vid körning? Notera att det faktiskt var Davids mening att avsluta programmet om ett exekveringsfel uppstår.
(1p)
- b) David kan enkelt åtgärda felet med en mindre rättelse på åtminstone två olika sätt. Redogör för vart och ett av dessa sätt genom att beskriva dem med vanliga ord, med pseudokod eller med faktiskt javakod.
(2p)
6. Beskriv huvuddragen i hur man parsar (tolkar) HTML-sidor med hjälp av inbyggda komponenter i Java. Du behöver inte ange alla detaljer, utan bara vilka objekt som är inblandade och en översiktlig beskrivning av hur det fungerar. Beskriv relationen mellan parsning av en HTML-sida och analys av de hyperlänkar som finns på sidan.
(3p)

7. Ange för vart och ett av följande påstående om det är sant eller falskt.

- 1) TCP är ett applikationsprotokoll.
- 2) PHP bygger liksom JSP på att skriptkod bäddas in i den vanliga HTML-koden med hjälp av särskilda HTML-taggar.
- 3) En datagramsocket av unicast-typ kan användas för att omväxlande sända datapaketer till olika mottagare.
- 4) Dataströmmen i TCP översätts till en följd av UDP-paket som skickas över nätverket och som sedan sätts ihop igen till en ström på mottagarsidan.

(2p)

8. En förbindelse för strömmande media kan tänkas använda sig av tre olika protokoll: RTP, RTCP och RTSP. Beskriv översiktligt vart och ett av dessa protokoll och deras relation till varandra.

(3p)

9. Betrakta följande kod och ange vilken typ av problem som skulle kunna uppstå vid exekvering. Ange villkoren för att den typen av problem ska kunna uppstå.

```
public class Tenta2008 {  
  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
  
        Tenta2008C1 o1 = new Tenta2008C1();  
        Tenta2008C2 o2 = new Tenta2008C2();  
  
        Tenta2008T1 t1 = new Tenta2008T1(o1, o2);  
        Tenta2008T2 t2 = new Tenta2008T2(o1, o2);  
        t1.start();  
        t2.start();  
    }  
}  
  
public class Tenta2008C1 {  
  
    synchronized void m1(Tenta2008C2 o2) {  
        System.out.println("C1 m1");  
        try {  
            Thread.sleep(1000);  
        } catch (Exception e) {  
        }  
        o2.m2();  
    }  
  
    synchronized void m2() {  
        System.out.println("C1 m2");  
    }  
}
```

Vänd, koden fortsätter på nästa sida.

```
public class Tenta2008C2 {

    synchronized void m1(Tenta2008C1 o1) {
        System.out.println("C2 m1");
        try {
            Thread.sleep(1000);
        } catch (Exception e) {
        }
        o1.m2();
    }

    synchronized void m2() {
        System.out.println("C2 m2");
    }
}

public class Tenta2008T1 extends Thread {

    Tenta2008C1 o1;
    Tenta2008C2 o2;

    Tenta2008T1(Tenta2008C1 o1, Tenta2008C2 o2) {
        this.o1 = o1;
        this.o2 = o2;
    }

    public void run() {
        o1.m1(o2);
    }
}

public class Tenta2008T2 extends Thread {

    Tenta2008C1 o1;
    Tenta2008C2 o2;

    Tenta2008T2(Tenta2008C1 o1, Tenta2008C2 o2) {
        this.o1 = o1;
        this.o2 = o2;
    }

    public void run() {
        o2.m1(o1);
    }
}
```

(2p)

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