

Tentamen

Nätverksprogrammering

Del 1

2018-08-23, 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. URLer och URler används för att identifiera resurser på nätet, t.ex. https://cs.lth.se/edaf65/foerelaesningar/?no_cache=1.
 - a) Vad är skillnaden på en URL och en URI?
(1p)
 - b) En URL kan delas upp i flera delar. Vad representerar de olika delarna?
(2p)
 2. I Javas trådmodell kan en tråd vara i någon av tillstånden *ready*, *running*, *blocked* eller *waiting*. I tillstånden *blocked* eller *waiting* sover tråden.
 - a) Vad får tråden att väckas från tillstånden *blocked* respektive *waiting*? Det finns tre scenarier.
(3p)
 - b) Kan en tråd äga en monitor (hålla monitorlåset) medans den sover? I så fall hur?
(1p)
 3. XML är ett dokumentformat som kan användas vid kommunikation mellan datorer.
 - a) Vad innebär det att ett XML-dokument är *valid* respektive *verified*?
(1p)
 - b) Vad är XPATH och vad används det till?
(1p)
 4. Utseendet på en hemsida kan styras med "style-direktiv", t.ex. `color: #F00` som anger röd färg. I uppgiften kan du utgå från html-koden `spaceman spiff was here` och beskriva tre olika sätt att ändra färgen på texten till röd.
 - a) Hur får man texten att bli röd enbart genom att ändra html-taggen ovan?
(1p)
-

- b) Layoutinformationen brukar samlas i en css-fil. Hur kopplar man html-taggar till en viss layout från css-filen? Ge ett exempel som inkluderar både texten i css-filen och html-taggen och ett exempel som bara involverar css-filen.

(2p)

5. Vilka av följande sex påståenden är korrekta? Rätt svar get +0,5p, felaktigt svar -1p. Lämnas inget svar ger det 0p. Hela uppgiften kan inte ge en negativt poäng.

- a) UDP garanterar inte att paketets innehåll är korrekt om det kommer fram.
- b) I ett XML-dokument kan man lagra binär data direkt (utan att omvandla den binära datan till text genom base64-kodning).
- c) Genom att använda nyckelordet `synchronized` i signaturen på en metod garanteras ömsesidig uteslutning mellan olika trådar i den metoden.
- d) I teckenkodningen UTF8 används 8 bitar för varje tecken.
- e) UDP-paket kan bara skickas till portar 1024 och uppåt.
- f) En html-fil kan innehålla JavaScript.

(3p)

6. Betrakta JavaScript-koden

```
function exam(){
  let student = { name: 'Kalle', age: 6 };
  if(student.alterEgo) {
    student.name = 'stupendous man';
    student.grade = '5';
  }
  if(student.age === 6){
    student.grade = 'u';
  }
  return student;
}
```

- a) I javascript finns jämförelseoperationerna `===` och `==`. Är det någon skillnad mellan dem? I så fall, vad skiljer?

(1p)

- b) Vad blir resultatet om koden körs, d.v.s. skrivs några felmeddelanden ut och vad är värdet på student-objektet som funktionen returnerar om exekveringen kommer så långt?

(2p)

7. Koden nedan körs hos en bank. Bankens system består flera trådar, som är instanser av klassen `Communication`. Varje tråd hantera en anslutning som kommer från t.ex. en bankomat, mobil-appen eller online-banken. Alla trådarna får samma `Bank`-objekt som parameter till sin konstruktor. Banken har noterat att kunder ibland är anslutna både i mobil-appen och online-banken samtidigt. Koden nedan fungerar inte alltid. Varför?

(2p)

```
public class Account {
    private String accountNbr;
    private double balance;

    synchronized public void setBalance(double balance) {
        this.balance = balance;
    }

    synchronized public double getBalance() {
        return balance;
    }
}

public class Communication extends thread {
    private Bank bank;

    public Communication(Bank bank){
        this.bank = bank;
    }

    public void run(){
        // The code for communication with ATM, mobile app and webb
        // eventually it will call withdraw()
    }

    private void withdraw(double amount, String accountNbr){
        Account account = bank.findAccount(accountNbr);
        while(bank.lock) { }
        bank.lock = true;
        account.setBalance(account.getBalance() - amount);
        bank.lock = false;
    }
}
```

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