



TENTAMEN

Luleå tekniska universitet

Kurskod: D0036D
Kursnamn: Nätverksprogrammering
Tentamensdatum: 2018-12-21
Skrivtid: 09.00-14.00
Tillåtna hjälpmedel: Inga

Jourhavande lärare m fullständigt telefonnr: Örjan Tjernström 0910-585716	Jourhavande lärare m fullständigt telefonnr:
Jourhavande lärare m fullständigt telefonnr:	Jourhavande lärare m fullständigt telefonnr:

Betygsgränser:	50% – betyg 3, bestäms senare – betyg 4, bestäms senare – betyg 5
Totalt antal uppgifter och poäng:	10 uppgifter, 40 poäng
Övriga upplysningar:	Jourhavande lärare är nåbar på telefon.

Allmänna anvisningar

Kontrollera att du fått samtliga uppgifter.
Besvara endast en uppgift per lösningsblad.
Skriv tydligt.

Efter tentamen

Tentamensresultatet syns ”Mina sidor” på Studentwebben.
Examinationsresultat ska meddelas inom 15 arbetsdagar efter. För kurser med fler än 60 examinerande samt för distansutbildningar får resultatet meddelas senast än 20 arbetsdagar efter.

Svaren på nedanstående uppgifter skall vara relaterade till aktuell kurs för att kunna ge poäng. Dina svar skall endast gå att tolka på ett sätt.

1. **Molntjänster:**

- a. Ni har fått ett uppdrag att utveckla en nätverksbaserat tjänst som för varje användare tar emot en fil, krypterar den och skickar en krypterad version tillbaka till användaren. Beskriv en arkitektur av tjänsten som är anpassat för att skaleras/skalas ut. (2p)
- b. Beskriv kortfattat funktionalitet av en lastbalanserare. (2p)

2. **Security & cheating**

- a. Nämn 4 metoder för att motverka fusk och säkerhetsproblem i ett nätverksprotokoll för ett spel. (2p)

3. **Nätverk och spel:(4p)**

- a. Vad är dead reckoning i sammanhang med spel och nätverk?
- b. Vilken typ av information/data skulle man använda för att implementera dead reckoning i ett biltävlingsspel?
- c. Vilket problem kan uppstå när man använder sig av dead reckoning och beskriv två metoder för att lösa det?

4. **URL:**

- a. Vilka delar består en URL av? (1p)

5. **Non-Blocking IO** är ett sätt att låta snabba CPU:er undvika blockerande operationer som väntar på ett långsamt nätverk.

- a. NIO använder channels. Hur fungerar dessa? (2p)

6. **Sockets:** Berkley Socket API:n innehåller en rad funktioner. Dessa används i olika kombinationer beroende på om det är TCP, UDP, Server, Klient. Vilka funktioner används, vad är deras uppgift, i vilken ordning används de om du skall implementera en: (5p)

- a. TCP server?
- b. UDP client?

7. **OSI:** är en lageruppdelad nätverksmodell där varje lager har sina specifika uppgifter. Vissa protokoll och teknologier mappar väldigt bra mot just det lager de ligger i med avseende på deras ansvar. De nedan uppräknade är sådana. Vilka är dessas absolut viktigaste uppgifter?(3p)

- a. IPv6
- b. Ethernet
- c. TCP

8. **Trådar:**

- a. Beskriv när synchronisering behövs.(2p)
- b. Förklara begreppet deadlock, när det uppstår.(1p)
- c. Beskriv hur man undviker deadlock.(1p)
- d. Beskriv utförligt tre olika sätt att "få" ett resultat från en tråd.(3p)

9. **Serialisering:**

- a. Vad innebär Serialisering?(1p)
- b. Ge exempel på en klass i Java som inte är serialiserbar och därför inte har implementerat Serializable?(1p)

10. **Applikationsprotokoll:** Ett väderlekssystem bestående av sensorer som levererar data till en väderleksserver. Klienter skall kunna: Registrera sig hos servern. Få en lista med serverns sensorer. Begära väderleksdata, från önskade sensorer, av servern. Avregistrera sig. Till detta behövs ett protokoll. I framtiden kan nya versioner av protokollet komma att skapas och servern ska då kunna hantera alla versioner. Protokollet skall vara språkneutralt.

- a. Ett antal meddelanden måste definieras i protokollet. Meddelandena skall vara i binärt format (ej klartext). Exempel på ett meddelande är *Unregister*. Detta skickas av klient för att avregistrera sig hos servern. Ge exempel på fyra meddelanden, utöver *Unregister*, samt vad dessa gör.(2p)
- b. En header med ett antal fält måste vara gemensamma för alla meddelanden. Vilka fält skall den innehålla och vad är syftet med dessa?(3p)
- c. Beskriv så att en, för dig okänd, programmerare kan implementera *Unregister*.(4p)
- d. Val av transportprotokoll är fritt men valet påverkar delar av designen för protokollet. Välj ett av transportprotokollen, motivera valet och beskriv hur det påverkar designen av protokollet.(1p)