**18-12-21**

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)

top-level domain(TLD) - är till för att fördela domäneradresser mellan länder på största skalan (t.ex. .uk, .dk, .no, .se), eller för reklam eller organisationer (.com, .org).

domain name - används för att underlätta sökandet. istället för att behöva skriva IP-addressen.

Protocol - Is http the protocol used when browsing the World-Wide Web (www) https is http with TSL (SSL) encryption.

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 APIn 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 – se till att alla noder har en unik adressering per entitet på nätverket.

b. Ethernet – att alla data kommer fram via kabel från routern/switchen till klienten

c. TCP – att avsändaren får svar från mottagaren om att paketet var mottaget.

8. **Trådar:**

a. Beskriv när syncronisering behövs.(2p)

b. Förklara begreppet deadlock, när det uppstår.(1p) - när två processer vill väntar på varandra i så att de kan börja använda motpartens resurs, men väntar indefinit eftersom båda har exklusivt ockuperat resurserna.

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)

**19-08-26**

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. Nätverk och spel:(3p)

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?

3. Nätverk och spel: För att implementera ett multiplayer spel över nätverk finns olika strategier.(4p)

d. Beskriva en möjligt implementation av en client-server arkitektur och av en peer to peer arkitektur.

e. Vilka för och nackdelar har dessa två?

2. **URL:**

a. Vilka delar består en URL av? (1p)

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

a. I stället för att skicka bytes använder NIO channels. Hur fungerar detta? (2p)

b. Vad gör man i stället för att blockera på read i en while loop? (2p)

4. **Sockets:** Socket-APIn innehåller en rad funktioner. Anta att det rör sig om en TCP socket. Beskriv följande funktioners uppgift. Samt koppla dem till TCP protokollet.(2p)

a. bind()

b. accept()

4. **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?(2p)

a. IPv6

b. Ethernet

5. **Trådar:**

a. Beskriv när syncronisering behövs.(2p) - för att undvika ett race condition

a. Förklara begreppet deadlock, när det uppstår.(1p)

b. Beskriv hur man undviker deadlock.(1p)

b. Beskriv utförligt tre olika sätt att ”få” ett resultat från en tråd.(3p)

6. **Serialisering:**

a. Vad innebär Serialisering?(1p)

b. Hur skiljer sig Serialisering åt mellan JAVA och C++?(2p) - java serieialisering ger gratis kompression vid serialisering. I Java är resultatobjektet är mindre än det ursprungliga objektet. I c++ behåller resultatobjektet samma storlek eller vid visa fall är det t o m större än original objektet efteråt. Vissa menar att i C++ är objekten redan något komprimerade från början, vilket gör det svårare att komprimera det ännu mer vid serialisering. (I java vinns mer “overhead” runt omkring det ursprungliga objektet). I java finns mellanrum vilket gör att objektet uppfattas större än vad den egentliga datamängden är.

7. **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. Tre fält måste vara gemensamma för alla meddelanden. Vad skall dessa fält innehålla?(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)

**19-10-26 med bilagor**

1.**Molntjänster**:

1. Ni har fått ett uppdrag att utveckla en nätverksbaserad 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)
2. Beskriv kortfattat funktionalitet av en lastbalanserare. (2p)

2.**Nätverk och spel:** Det är av största vikt att hålla trafikmängd på så låg nivå som möjligt när det handlar om datorspel över nätverk.

1. Beskriv två olika tillvägagångsätt för att genomföra detta. (2p)
2. Beskriv nackdelar med de båda sätten. (2p)

3.**Nätverk och spel:** Massively multiplayer online (MMO) behöver använda sig av ett nätverk av Servrar (Server-network) för att dela arbetsbördan. Två angreppssätt föruppdelningen har belysts i kursen. (4p)

1. Beskriva dessa.
2. Vilka för och nackdelar har dessa två?

4.**Sockets:** Socket-APIn innehåller en rad funktioner. Anta att det rör sig om en TCPsocket. Beskriv följande funktioners uppgift. Samt koppla dem till TCP-protokollet. (2p)

1. bind()
2. accept()

5.**OSI:** är en lageruppdelad nätverksmodell där varje lager har sina specifikauppgifter. 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)

1. IPv6
2. Ethernet
3. TCP

6.**Trådar:**

1. Beskriv begreppet race conditions.(1p)
2. Beskriv begreppet data race. (1p)
3. Beskriv hur man undviker deadlock.(1p)

7.**Serialisering:** I både Java och C++ har vi en klass som med en medlem som skapatsmed new operatorn.

1. Beskriv hur serialiseringen, för denna klass, går till i Java(2p)
2. Beskriv hur serialiseringen, för denna klass, går till i C++?(2p)

8.**Serialisering:** Host1 skickar en struct, bestående av 3 st unsigned integers, tillHost2. Det visar sig att structen tolkas fel av Host2. Vissa värden blir rätt men fler talet blir felaktiga (se Bilaga1 Tabell 1). De viktiga delarna av koden för Host1och Host2 finns i Bilaga 1.

1. Vad beror det på att det blir fel? (2p)
2. Åtgärda i koden så att det blir rätt. (4p)

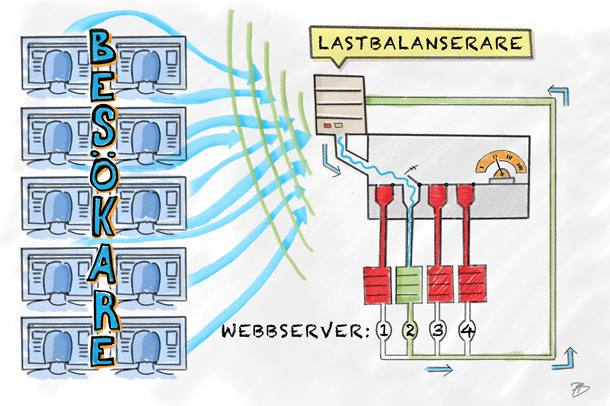
9.**Applikationsprotokoll:** Du har implementerat en spelplan som en server. Spelplanen består av på 101\*101 rutor. Rutorna kan ha tjugo färgnyanser eller vara utan färg. Din spelplan skall vara tillgänglig för flera klienter. UDP används för att kommunicera med din spelplan. Spelplanen skall hantera upp till och med tjugo samtidiga klienter. Varje klient har sin egen färg. Klienterna har rätt attantingen skriva sin färg till en tom ruta eller radera sin egen färg från en ruta. Din spelplan kontrollerar att detta följs. Det är av största vikt att klienterna får veta om deras kommando kunde utföras eller inte. För att kommunicera med din spelplan behövs ett protokoll. I framtiden kan nya versioner av protokollet komma att skapas och spelplanen ska då kunna hantera alla versioner. Din uppgift är att konstruera ett protokoll som sköter allt detta. Ett antal meddelanden måste definieras i protokollet. Meddelandena skall vara i binärt format (ej klartext). Protokollet skall vara språkneutralt. Beskriv så att en, för dig okänd, programmerare kan implementera ditt protokoll. (10p)

1. Vilka meddelanden har ditt protokoll?
2. Hur ser meddelandena ut?
3. Vilket syfte har de?

**19-10-26**

1. **Molntjänster:**

a. Ni har fått ett uppdrag att utveckla en nätverksbaserad 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) - 

## **Lastbalansering fördelar trafiken för att skapa en mer tillgänglig webbsida**

Enkelt uttryckt kan man säga att lastbalansering gör så att din webbtjänst finns tillgänglig på Internet i högre grad. Metoden innebär att trafik delas upp mellan olika resurser.

En vanlig lösning är att en tjänst först tar emot inkommande trafik, för att sedan vidarebefordra trafiken till en av flera webbservrar som finns i bakgrunden. Webbservern skickar i sin tur trafiken vidare tillbaka via tjänsten. Syftet är att belasta de webbservrar som har lite att göra, så att trafiken inte “svämmar över” och skapar problem. Tekniken kallas för multiplexering och innebär just att flera användare eller processer delar på samma resurs. På engelska använder man begreppet Load Balancing när man talar om lastbalansering. Round Robin DNS och NLB är alternativa metoder som tidigare använts, men övergivits, då ny teknik varit effektivare.

2. **Nätverk och spel:**(3p)

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?

3. **Nätverk och spel:** Massively multiplayer online (MMO) behöver använda sig av ett nätverk av Servrar (Server-network) för att dela arbetsbördan. Två angreppssätt för uppdelningen har belysts i kursen. (4p)

a. Beskriva dessa.

b. Vilka för och nackdelar har dessa två?

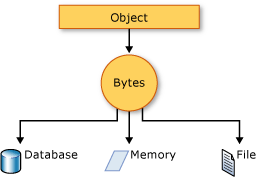
4. **Sockets:** Berkeley socket API innehåller en rad funktioner, vissa används till TCP andra till UDP och några till både TCP och UDP. Vilka funktioner, från Berkeley socket API, används och i vilken ordning används de om du ska implementera en

a. TCP-client? (3p)

b. UDP-server? (3p)

5. **Serialisering:**

a. Vad innebär Serialisering?(1p)



**Serialization** is the process of converting an object into a stream of bytes to store the object or transmit it to memory, a database, or a file. Its main purpose is to save the state of an object in order to be able to recreate it when needed. The reverse process is called deserialization.

b. Vad gör seariliseringen så mycket lättare i Java jämfört med C eller C++?(1p)

6. **Multi Client Server:**

a. Du ska implementera en TCP server som ska betjäna flera anslutna clienter samtidigt. Beskriv hur du löser detta. Använd Sekvensdiagram, kod eller liknande i din beskrivning.(4p)

7. **Trådar:**

a. Beskriv begreppet race conditions.(1p)

b. Beskriv begreppet data race.(1p)

8. **Endianes:** Host1 skickar en struct, bestående av 3 st 32-bit unsigned integers, till Host2. Host1 körs på en maskin som använder byteordningen Little endian. Host2 körs på en maskin som använder byteordningen Big endian. De viktiga delarna av koden för Host1 och Host2 finns i Bilaga 1.

a. Om structens värden hos Host1 är x=4 278 190 080 , y=268 505 089, z=16 000 000, vilka värden kommer då Host2 att ha i sin struct efter att överföringen slutförts? Svara med decimala eller hexadecimala värden. (se Tabell 1 i Bilaga 1) (2p)

b. Åtgärda i koden så att värdena hos Host2 överensstämmer med de som Host1 överför. (2p)

9. **URL:**

a. Vilka delar består en URL av? (1p)

10. **Applikationsprotokoll:** Du har implementerat en spelplan som en server. Spelplanen består av på 101\*101 rutor. Rutorna kan ha tjugo färgnyanser eller vara utan färg. Din spelplan skall vara tillgänglig för flera klienter. UDP används för att kommunicera med din spelplan. Spelplanen skall hantera upp till och med tjugo samtidiga klienter. Varje klient har sin egen färg. Klienterna har rätt att antingen skriva sin färg till en tom ruta eller radera sin egen färg från en ruta. Din spelplan kontrollerar att detta följs. Det är av största vikt att klienterna får veta om deras kommando kunde utföras eller inte. För att kommunicera med din spelplan behövs ett protokoll. I framtiden kan nya versioner av protokollet komma att skapas och spelplanen ska då kunna hantera alla versioner. Din uppgift är att konstruera ett protokoll som klarar detta. Ett antal meddelanden måste definieras i protokollet. Meddelandena skall vara i binärt format (ej klartext). Protokollet skall vara språkneutralt. Fält som förekommer i alla meddelanden kan ses som en header. Beskriv så att en, för dig okänd, programmerare kan implementera ditt protokoll. Till din hjälp har du nedanstående frågor. (10p)

a. Vilka meddelanden har ditt protokoll?

b. Vilka syften har protokollets meddelanden.

c. Beskriv hur headern ser ut?

d. Beskriv hur resten av respektive meddelande ser ut?