

Kursnamn: Nätverksprogrammering	Kurskod: D0036D
Antal timmar: 5	Datum: 20191026

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. Gränser för betyg 4 och 5 bestämmes senare.	Antal uppgifter och totalpoäng: 9 40
---	---

Övriga uppgifter: Elektronisk utrustning skall vara avstängd under tentamen. Svaren på nedanstående uppgifter skall vara relaterade till aktuell kurs för att kunna ge poäng.
--

Allmänna anvisningar

Kontrollera att du fått samtliga uppgifter.
Besvara endast en uppgift per lösningsblad. Skriv inte på baksidan.
Skriv tydligt, texta gärna och använd inte röd penna.

Efter tentamen

Tentamensresultat meddelas senast tre veckor efter tentamenstillfället och senast två veckor före nästa omtentamenstillfälle.
Tentamensresultatet syns på Mitt LTU – Ladok för studenter.
Din rättade tentamen skannas och blir synlig på Mitt LTU – Rättade tentor.

Uppgifter till tryckeriet

Projektnummer: 341980	Antal exemplar 26	Antal sidor: 4
Övriga upplysningar: :		

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. **Nätverk och spel:** Det är av största vikt att hålla trafikmängd på så låg nivå som möjligt när det handlar om datorspel över nätverk.
 - a. Beskriv två olika tillvägagångssätt för att genomföra detta.(2p)
 - b. 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 för uppdelningen har belysts i kursen. (4p)
 - a. Beskriva dessa.
 - b. Vilka för och nackdelar har dessa två?
4. **Sockets:** Socket-API:n 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()
5. **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
6. **Trådar:**
 - a. Beskriv begreppet race conditions.(1p)
 - b. Beskriv begreppet data race.(1p)
 - c. Beskriv hur man undviker deadlock.(1p)
7. **Serialisering:** I både Java och C++ har vi en klass som med en medlem som skapats med new operatören.
 - a. Beskriv hur serialiseringen, för denna klass, går till i Java(2p)
 - b. Beskriv hur serialiseringen, för denna klass, går till i C++?(2p)
8. **Serialisering:** Host1 skickar en struct, bestående av 3 st unsigned integers, till Host2. Det visar sig att structen tolkas fel av Host2. Vissa värden blir rätt men flertalet blir felaktiga (se Bilaga1 Tabell 1). De viktiga delarna av koden för Host1 och Host2 finns i Bilaga 1.
 - a. Vad beror det på att det blir fel? (2p)
 - b. 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 simultana 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 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)
- Vilka meddelanden har ditt protokoll?
 - Hur ser meddelandena ut?
 - Vilket syfte har de?

Bilaga 1

Värden från Host1	Tolkad värden hos Host2
0	0
1	16777216
65792	65792
16000000	2421760
16776960	16776960
67108864	4
4294967295	4294967295

Tabell 1

```
/*
Struct Test finns hos både Host1 och Host2.
*/

struct Test
{
    unsigned int x;
    unsigned int y;
    unsigned int z;
};

/*
Host1 kod
*/
Test test;

/*
För uppgiften oviktig kod bortkommenterad.
*/

int size = sizeof Test;
char* buff = new char[size];
memcpy((void*)buff, (void*)&test, sizeof Test);

/*
Här är koden, som ska skicka
innehållet i buff till Host2, bortkommenterad.
*/

/*
Host1 kod slut.
*/

/*
Host2 kod
*/

Test test;
int size = sizeof Test;
char* buff = new char[size];

/*
Här är koden, som tar emot trafik från Host1 och lägger
innehållet i buff, bortkommenterad.
*/

memcpy((void*)&test, (void*)buff, sizeof Test);

/*
Host2 kod slut.
*/
```