

72 timers HJEMMEEKSAMEN - KONTE

PG3401 C Programmering

Tillatte hjelpemidler: Alle

Varighet: 72 timer

Karakterskala/vurderingsform: Nasjonal karakterskala A - F

Dato: 2. august - 5. august 2024

Oppgavesettet har 4 sider. Det er totalt 4 oppgaver i oppgavesettet.

Det er 72 timers frist på denne hjemmeeksamen. Vær obs på at eksamen MÅ leveres innen fristen som er satt i Wiseflow, og oppgaven kan kun leveres via WISEFLOW. Det vil ikke være mulig å få levert oppgaven etter fristen – det betyr at du bør levere i god tid slik at du kan ta kontakt med eksamenskontoret eller brukerstøtte hvis du har tekniske problemer.

Det presiseres at studenten skal besvare eksamen selvstendig og individuelt, samarbeid mellom studenter og plagiat er ikke tillatt. Det er ikke tillatt å presentere andres arbeid som ditt eget – dette inkluderer arbeid utført av kunstig intelligens (som tekst- eller kode-genereringsmodeller). Eksamen skal løses på Linux.

Merk at oppgavene er laget med stigende vanskelighetsgrad, og spesielt de to siste oppgavene er vanskeligere enn de første oppgavene. Det oppfordres derfor til å gjøre de første oppgavene (helt) ferdige slik at studenten ikke bruker opp all tid på å gjøre de siste oppgavene først.

Format på innlevering

Dette er en praktisk programmeringseksamen (bortsett fra oppgave 1), fokus bør derfor være på å forklare hvordan du har gått frem, begrunne valg og legge frem eventuelle antagelser du har gjort i din løsning.

Hvis du ikke klarer å løse en oppgave er det bedre om du forklarer hvordan du har gått frem og hva du ikke fikk til – enn å ikke besvare oppgaven i det hele tatt. Det forventes at alt virker hvis ikke annet er beskrevet i tekstbesvarelsen, hvis du vet at programmet krasjer, ikke kompilerer eller ikke virket slik den var tenkt er det viktig å forklare dette sammen med hvilke skritt du har tatt for å forsøke å løse problemet.

Besvarelsen skal leveres inn i PDF format i en fil med navn PG3401_konte24_[kandidatnummer].pdf. Kildekoden må være vedlagt i 1 ZIP fil, navnet på filen skal være PG3401_konte24_[kandidatnummer].zip. Denne filen skal ha følgende struktur:

```
\ oppgave_2 \ makefile
\ oppgave_2 \ [...]
\ oppgave_3 \ makefile
```

```
\ oppgave_3 \ [...]  
\ oppgave_4 \ makefile  
\ oppgave_4 \ [...]
```

Vær sikker på at alle filer er med i ZIP filen. Hver mappe skal ha en makefile fil, og det skal ikke være nødvendig med noen endringer, tredjeparts komponenter eller parametere – sensor vil i shell på Debian Linux 10 gå inn i mappen og skrive «make» og dette skal bygge programmet med GCC.

Tekstbesvarelsen skal inneholde besvarelse på oppgave 1 (skriv det kort og konsist, trenger ikke noe stor avhandling). Etter den rene tekstbesvarelsen skal hver oppgave inkludere begrunnelse/dokumentasjon til kildekoden, hver av disse begrunnelsene skal være på en ny side for å gjøre tekstbesvarelsen oversiktlig for sensor. Besvarelser i andre formater (enn PDF og ZIP) vil ikke bli lest.

Det spesifiseres at oppgaven må programmeres på Linux (tekstbesvarelsen kan skrives på Mac eller Windows).

Oppgave 1. Generelt (5 %)

- a) Forklar hva C programmeringsspråket kan brukes til.
- b) Hvem er Dennis Ritchie og hva er han kjent for innen Informasjonsteknologi?
- c) Forklar hva kommandoen «sudo» gjør på Linux.

Oppgave 2. Filer og funksjoner (25 %)

Last ned følgende kildefiler:

```
http://www.eastwill.no/pg3401/konte_h23_oppgave2_fib.c  
http://www.eastwill.no/pg3401/konte_h23_oppgave2_prim.c  
http://www.eastwill.no/pg3401/konte_h23_oppgave2_kvad.c  
http://www.eastwill.no/pg3401/konte_h23_oppgave2_perf.c  
http://www.eastwill.no/pg3401/konte_h23_oppgave2_abun.c  
http://www.eastwill.no/pg3401/konte_h23_oppgave2_def.c  
http://www.eastwill.no/pg3401/konte_h23_oppgave2_odd.c
```

Du skal skrive et program som tar ett og ett tall som input fra brukeren, for hver av tallene skal du kalle funksjonene i de overnevnte kildefilene for å finne ut om tallene er a) et tall i Fibonacci-rekken, b) et primtall, c) et kvadrattall, d) et perfekt tall, e) et overflødig tall, f) et defekt tall, eller g) et oddetall. Skriv ut i terminalen hvilket tall som er behandlet, og om tallet er en (eller flere) av de forskjellige tall «typene».

Programmet skal så lage en struct som inneholder en int som beskriver tallet brukeren tastet inn som input – og 7 verdier (som int eller som bool) som beskriver resultatet fra å kalle funksjonene A til G over. Denne structen skal skrives BINÆRT til en output fil.

Du må lage en makefile fil (hentet fra forelesning 4, sleide 63), og en header fil til hver av de nedlastede kildefilene for programmet til å kompilere riktig, gjør eventuelle endringer påkrevet i makefile filen og kildefilene for å få programmet til å bygge og kjøre korrekt.

Dokumentasjonskrav:

Utdatafilen skal ligge i samme mappe som programmet, og utdata filen må være en del av den innleverte besvarelsen. Du skal i tillegg ta skjermbildet av at du kjører programmet, dette skal både legges ved som PNG fil levert sammen med kildekoden, og skal også settes inn som bilde i PDF besvarelsen.

Oppgave 3. Nettverk (35 %)

Du skal skrive to programmer som kommuniserer med hverandre ved hjelp av en implementasjon av FTP – File Transport Protocol, hvor ditt klient-program skal kunne laste opp og laste ned filer fra server-programmet. Server-programmet skal ta et argument på kommandolinje som sier hvilken port den skal lytte på, klient-programmet skal ta argument(er) på kommandolinje som sier hvilken IP adresse og portnummer den skal koble seg til.

Last ned filen

http://www.eastwill.no/pg3401/konte_h24_oppgave3.c

Denne filen inneholder en funksjon for autentisering som skal brukes av server programmet. (Denne filen leser en fil som heter user.db, denne filen er forventet å inneholde brukernavn og passord på etterfølgende linjer med semikolon mellom – les koden for å se formatet på filen og opprett denne filen for din test.) Resten av koden for både klient og server må du lage selv.

Du må lage to makefile filer (hentet fra forelesning 4, sleide 63), en for server-programmet og en for klient-programmet. Du skal også lage en header fil for C filen du lastet ned, gjør eventuelle endringer påkrevet i makefile filen og kildefilen for å få programmet til å bygge og kjøre korrekt.

Ditt program skal som minimum støtte følgende FTP kommandoer:

PUT; kopierer en fil fra klient til server

GET; kopierer en fil fra server til klient

Dokumentasjonskrav:

I tillegg til å levere inn kildekoden og makefile filene du har laget skal du også levere inn filen user.db som du har testet med. Du skal så ta skjermbilde av kjøring av programmet, både av klient- og server-programmet, og disse skal både legges ved som PNG filer levert sammen med kildekoden, og skal også settes inn som bilder i PDF besvarelsen.

Oppgave 4. Tråder (35 %)

Du skal lage et program som leser en fil, dette programmet skal ha 10 arbeidstråder som er ansvarlig for å lese filen. Hver arbeidstråd skal ha følgende funksjonalitet:

- Lese 4096 bytes fra filen
- I en loop summere sammen hver byte som er lest i en 32 bits integer som vil holde en løpende sum
- Regne ut summen modulo 2000
- Bruke funksjonen «sleep» for å vente i så mange millisekunder som svaret
- Printe ut på skjerm a) hvilken tråd som printer ut, for eksempel i form av en ID du selv tilordner hver tråd, og b) hvor mange bytes som ble lest, hva summen ble, og hva modulo ble

Programmet ditt må i tillegg ha følgende funksjonalitet/kvaliteter:

- Lag en makefile fil (hentet fra forelesning 4, sleide 63) for å programmet til å bygge, gjør eventuelle endringer påkrevet i makefile filen for å få programmet til å bygge og kjøre korrekt
- Bruke C språkets primitiver for synkronisering mellom trådene
- Legg til kommentarer i koden for å dokumentere hva koden gjør

Dokumentasjonskrav:

Du skal teste programmet ditt med denne PDF filen som inneholder oppgaveteksten for eksamen, du skal vise et skjermbilde fra når programmet kjører som viser flere forskjellige tråder som printer ut tekst på skjermen. Skjermbildet skal både legges ved som PNG filer levert sammen med kildekoden, og skal også settes inn som bilder i PDF besvarelsen.

+

Slutt på oppgavesettet