

All kode skal være lett å lese og med gode kommentarer. Alt kjører og gjør som det skal etter mine tolkninger av oppgavene.

Oppgave 1:

- a) Forklar hva C programmeringsspråket kan brukes til.

C kan brukes til ekstremt mye og er svært populært selv om det er et relativt gammelt språk. Styrken med å bruke C er at man som programmerer har full kontroll over hva som skjer og det er utrolig raskt.

Det er svært populært blant annet innenfor IoT / embedded, operativsystemer, drivere og kompilatorer.

- b) Hvem er Linus Torvalds og hva er han kjent for innen informasjonsteknologi?

Linus Torvalds er en finsk programmerer som studerte computer science. Han er mest kjent for å stå bak operativsystemet Linux (som er oppkalt etter hans navn). Han er også kjent for å stå bak det populære versjonskontrollsystemet "git".

- c) Hvis du laster ned en binær eksekverbar fil fra internett kan det være at filen ikke kan kjøres fordi den ikke har "execute" rettigheter satt. Forklar hvordan du kan løse dette på kommandolinje i Linux.

For å gjøre endringer i en fil eller mappes rettigheter bruker man kommandoen "chmod" i terminalen. For å gi en fil "execute" rettigheter skriver man "chmod +x filnavn" hvor "x" står for "execute". Samme vil være for å fjerne rettigheter bare med "-" i stedet for "+".

Oppgave 2:

- a) Jeg har tatt utgangspunkt i kode jeg har laget i oppgaver i timen for lesing av fil. Jeg åpner en fil "data.dta" i lesemodus med den enkodede dataen vi fikk oppgitt i oppgaven. Ettersom hex oppgis som to og to tegn vil jeg altså loope to og to tegn av gangen.

For å konvertere fra hex til tall som kan representeres i ASCII tabellen vil man først få konvertert tallene/bokstavene i hex til tall (f.eks A = 10, B = 11 osv) hver for seg.

Disse tallene representeres med 4 bits hver (siden det går til max 15), så vi gjør en left shift med 4 på det første tallet (vil være det samme som å gange det første tallet med 16) for å så plusse de to tallene sammen. Slik får man tall tilbake hvor jeg velger å tolke det som karakterer når jeg skriver til fil.

- b) Ettersom oppgaven spesifiserer at det skal lages en funksjon som tar inn den dekodete teksten som parameter har jeg både skrevet den til fil og lagret den i et buffer i main metoden. Slik kan jeg kalle på funksjonen som skal printe alle forekomster per bokstav i alfabetet med bufferet med den dekodete dataen.

Logikken her vil være å loope gjennom et array med alfabetet og for hver bokstav sammenligne med hele det dekodete bufferet, for så å inkrementere en counter. Slik ser man hvor mange forekomster det er per bokstav i alfabetet.

Oppgave 3:

Her har jeg tatt utgangspunkt i kode fra oppgave jeg har løst i timen om linked list.

Jeg har tolket det slik at en bruker kan legge til flere “varer” i en liste og få kvitteringen printet ut. Jeg har valgt å gjøre det slik at hvis det tastes det samme varenavnet (uten å tenke på casing) flere ganger og prisen per vare er lik, så regner jeg det som samme vare og legger bare til antallet som er tastet inn på den varen som allerede er lagt til i listen. Dersom varenavnet er likt men prisen per vare er ulik, regner jeg det som en ny vare i listen. Dette kan f.eks være epler, hvor det er forskjellig type epler til forskjellig pris. Dette gjør også at delete funksjonaliteten fungerer som oppgitt i oppgaven at den skal slette alle varer i listen med samme navn.

Ellers skal alt kjøre og gjøre som oppgitt i oppgaven.

Oppgave 4:

Feil 1: I andre if setning i koden vi har fått settes `httpCode = 200`. Dermed vil det alltid gi true og derfor kan det gi feil informasjon ved at `blsSuccess` alltid vil være true. Her skulle det selvfølgelig stått `httpCode == 200`, med to likhetstegn.

Feil 2: I kodebiten som skal hente ut `content-length` gjøres det en `atoi()` operasjon, men på feil variabel. Det blir prøvd å utføre på hele http responsen. Her skulle det vært gjort på `pszPtr` istedet som peker til `content-length` på dette stadiet i koden.

Feil 3: I kodebiten hvor det forsøkes å hente ut servernavnet fra responsen vil det bli buffer overflow dersom servernavnet har mer enn 15 karakterer i seg. En fiks her vil være å bruke `strncpy` istedet for `strcpy` og oppgi hvor mange bytes som skal kopieres. Når jeg da skal sette tilbake `"\n"` så er det `pszPtr` som inneholder riktig lengde for beregning av posisjonen for denne. Siden den strengen man kopierer til kan bli kortere enn den man kopierer fra så må man bruke lengden på strengen som det kopieres fra for å beregne hvor `"\n"` skal skrives tilbake.

Dersom man ikke vil ha en "cap" på hvor mange bytes servernavn og content-type kan være, bør man heller allokere størrelsen i structen etter størrelsen på teksten i responsen og dermed kopiere det riktige antall bytes for å få med hele teksten. Da må man også selvsagt huske på å frigjøre den allokerne minneplassen. Dette vil også medføre at størrelsen på forskjellige instanser av structen kan variere.

Oppgave 5:

I denne oppgaven har jeg valgt å bruke semaphores som synkroniseringsteknikk. Jeg følte dette var et bedre valg fordi f.eks mutex som vi også har hatt om i timen brukes gjerne mer som en lock, mens semaphorer kan gi signaler når noe skal gjøres.

Jeg har tolket det slik at hvis brukeren skriver inn mer enn 10 tegn skal det skrives 10 og 10 inn i filen av gangen fra arbeidstråden og “\n” når hele setningen er skrevet inn. “\n” kunne vært håndtert i min kode ved å bare signalisere til arbeidstråden at det skal legges på en “\n” når alle tegn i bufferet fra userinput er skrevet over, men jeg brukte bare fgets() i stedet som gir en “\n” automatisk.

Oppgave 6:

For å finne ip-adressen til serveren oppgitt i oppgaven brukte jeg nslookup.

Jeg åpner en socket connection mot serveren vi har fått oppgitt i oppgaven. Deretter sendes det en GET message som ber om å få den spesifiserte html siden. Jeg har løst lesing av responsen slik at den bare leser et fast sett med bytes om gangen i et buffer helt til responsen er ferdiglest. Slik har jeg gjort koden mer robust.

Som spesifisert i oppgaven skulle dette være en "browser" så jeg tolket dette som at http response headers ikke skal inkluderes i utskriften i konsoll. Derfor har jeg laget en funksjon for å bare lese ut body (html siden) fra responsen.