

Obligatorisk oppgave i INF1060: Programmering i C

Utlevering: onsdag 9. september 2015

Innlevering: onsdag 23. september 2015 før kl. 23.59

Intro

Målet med denne oppgaven er å få trening i C-programmering før man begynner på hjemmeeksamenene som hovedsaklig vil være å løse programmeringsoppgaver i C. For å kompilere koden skal dere lage en Makefile (se man make og [HTML manualen](#)) slik at man bare kan skrive make i katalogen hvor filene er lagret for å kompilere programmet.

Programmet skal kunne kjøres på IFI's standard Linux-maskiner eller på Linux-klusteret. For å komme inn på en Linux-maskin, kan man f.eks. kjøre `ssh linux`.

Denne obligatoriske oppgaven skal *løses individuelt* og *må godkjennes av gruppelærer* før man kan gå opp til avsluttende eksamen. Det er ikke lov å hente kode fra annet hold eller dele egen kode med andre - all kode må være skrevet selv. Det er tillatt og vi oppfordrer sterkt til å diskutere problemer samt å utveksle informasjon og kunnskap med andre studenter, men altså ikke dele kode direkte. Se også på siste avsnitt om hvordan oppgaven skal leveres inn.

Dersom du får problemer, så kan du få [hjelp](#) hos Oraklene. Det er stor pågang rett før innleveringsfristen, så det lønner seg å være tidlig ute.

Det er to oppgaver som skal løses:

- [Oppgave 1 - Tekststrenger](#)
- [Oppgave 2 - Bitfikling/Kompressjon](#)

Oppgave 1 - Tekststrenger

Denne oppgaven består i å skrive et C-program som kan leke litt med tekststrenger. Et scenario kan for eksempel være barnesangen *Tre små kinesere på Høybro plass* hvor man bytter vokaler. Kort sagt skal programmet kunne utføre følgende oppgaver (i subseksjonene under er det spesifisert i mer detalj

hva programmet skal gjøre):

USAGE: ./a.out command input_file

where "command" is one of the following:

print	print input_file
random	print a random line
replace	replace the vowels with all the other vowels
remove	remove vowels
len	print the number of characters in the input_file

En eksempel fil (*tresmaa.txt*) ligger [her](#), og det er denne som er brukt i eksemplene under.

Håndtere input-parametere

Som vist over skal programmet kunne ta parametere som gir instruksjoner om hva som skal gjøres og hvilken tekstfil som skal brukes som input.

Legge tekstlinjer inn i en lenket liste

Filen som leses inn skal legges inn i en enkelt-linket liste med et element for hver linje i filen. Hvert element i listen skal bare inneholde en 'neste'-peker og en peker til en tekststreng. Første element i listen skal inneholde første linje i filen osv. Det siste som skal gjøres av programmet før det avslutter er å frigjøre de ressursene som er brukt i den linkede listen.

Kommandoer

De forskjellige kommandoene som skal støttes er de følgende:

print

hvis denne kommandoen er gitt skal programmet traversere listen og skrive ut teksten linje for linje. Det vil si at man bare skriver ut hele input-filen (etter å ha lagt den inn i en lenket liste). Et eksempel på utskrift kan være:

```
paalh@linux:~/inf1060/oblig1 > ./a.out print tresmaa.txt
```

Print text:

```
Tre små kinesere på Høybro plass  
satt og spilte på en kontrabass.  
Så kom en konstabel, spurte hva var hendt,  
tre små kinesere på Høybro plass.
```

random

denne skal skrive ut en tilfeldig linje i listen. Et eksempel på utskrift kan være:

```
paalh@linux:~/inf1060/oblig1 > ./a.out random tresmaa.txt
```

```
Print a random line  
satt og spilte på en kontrabass.
```

replace

denne skal medføre at programmet går i en løkke - en gang for hver vokal. For hver runde skal programmet sjekke tegn for tegn om det er en vokal eller ikke - hvis det er en vokal skal man bytte denne ut med den tilsvarende vokalen for denne runden. Skriv ut teksten for for hver runde. Et eksempel på utskrift kan være:

```
paalh@linux:~/inf1060/oblig1 > ./a.out replace tresmaa.txt
```

```
Replace vowels ...
```

```
... with vowel 'a'  
Tra sma kanasara pa Haabraplass  
satt ag spalta pa an kantrabass.  
Sa kam an kanstabal, sparta hva var handt,  
tra sma kanasara pa Haabraplass.
```

```
... with vowel 'e'  
Tre sme kenesere pe Heebrepless  
sett eg spelte pe en kentrebess.  
Se kem en kenstebel, sperte hve ver hendt,  
tre sme kenesere pe Heebrepless.
```

```
[...SNIP... - removed 'i', 'o', 'u', 'y' and 'æ']
```

```
... with vowel 'ø'  
Trø smø kønøsørø pø Høøbrøpløss  
søtt øg spøltø pø øn køntrøbøss.  
Sø køm øn kønstøbøl, spørtø hvø vør høndt,  
trø smø kønøsørø pø Høøbrøpløss.
```

```
... with vowel 'å'  
Trå små kånåsårå på Hååbråplåss  
sått åg spåltå på ån kåntråbåss.  
Så kåm ån kånståbål, spårtå hvå vår håndt,  
trå små kånåsårå på Hååbråplåss.
```

remove

programmet skal sjekke tegn for tegn om det er en vokal eller ikke - hvis det er en vokal skal denne fjernes (ikke byttes ut med blank) slik at det ikke skrives ut noen vokaler. Et eksempel på utskrift kan være:

```
paalh@linux:~/inf1060/oblig1 > ./a.out remove tresmaa.txt
```

```
Remove vowels:
```

```
Tr sm knsr p Hbrplss  
stt g splt p n kntrbss.  
S km n knstbl, sprt hv vr hndt,  
tr sm knsr p Hbrplss.
```

len

programmet skal skrive ut hvor mange tegn det er tilsammen i fila. Et eksempel på utskrift kan være:

```
paalh@linux:~/inf1060/oblig1 > ./a.out len tresmaa.txt
```

```
The text is 141 characters long
```

Oppgave 2

Denne oppgaven går ut på å lage et program som opererer på bit-nivå. Scenarioet er systemer som kanskje har begrenset med minne og ikke trenger så mange forskjellige tegn. Man kan da spare plass ved å bytte ut den standard ASCII-representasjonen av tegnene med noe som bruker færre bit per tegn. Vi skal i denne oppgaven lagre bilder og anta at vi bare trenger fire forskjellige tegn for å lagre et ASCII bilde. Disse kan da representeres med bare to bit hver (som igjen vil si at en byte inneholder 4 tegn):

tegn	bit-representasjon
' ' (blank)	00
:	01
@	10
\n (linjeskift)	11

Kort sagt skal programmet kunne utføre følgende oppgaver (i subseksjonene under er det spesifisert i mer detalj hva programmet skal gjøre):

```
USAGE: ./a.out command input_file output_file
```

where "command" is one of the following:

```
p    print input_file  
      (output_file is ignored if specified)  
e    encode/compress input_file to output_file  
d    decode/uncompress and print input_file  
      (output_file is ignored if specified)
```

Håndtere input-parametere

Som vist over skal programmet kunne ta parametere som gir instruksjoner om hva som skal gjøres og hvilken tekstfil som skal brukes som input og eventuelt output ved kommandoen 'e'.

Kommandoer

De forskjellige kommandoene som skal støttes er de følgende:

p

hvis denne kommandoen er gitt skal programmet skrive ut ASCII-bildet som er lagret i input-filen.

e

hvis denne kommandoen er gitt skal programmet lese inn input-filen tegn for tegn og komprimere tegnet til 2-bit-verdien beskrevet over. Verdiene skal så skrives ut igjen til output-filen (denne skal kunne leses inn igjen i dekomprimeringsoperasjonen).

d

hvis denne kommandoen er gitt skal programmet lese inn input-filen som er 'komprimert' og skrive ut det originale ASCII-bildet.

Vedlegg

Eksempelfiler ligger [her](#). Her er *.txt originale tekstfiler med 8-bits ASCII-representasjon og *.compressed er ferdig komprimerte filer med 2-bits representasjon av tegnene.

Innlevering

Besvarelsen skal bestå av den *godt kommenterte* kildekoden til programmet og en makefile for kompilering. Under finner dere instruksjoner på hvordan dere skal levere inn, og **ved å levere inn oppgaven samtykker dere på at dere overholder reglene for innleveringen.**

Elektronisk innlevering

Oppgaven skal leveres elektronisk, dvs. at ingen papirkopi er nødvendig. Dere skal opprette en katalog med deres *brukernavn som navn*. Denne katalogen skal inneholde alle filer som skal leveres (kode og makefile). Denne katalogen lager dere en tar-ball (tar) av og komprimerer med gzip. Tar-ballen skal navngis med

brukernavnet og med postfix .tgz - for eksempel *paalh.tgz* - og lastes opp på [Devilry](#) før tidsfristen utløper.

Regler for innlevering og bruk av kode

Ved alle pålagte innleveringer av oppgaver ved Ifi enten det dreier seg om obligatoriske oppgaver, hjemmeeksamen eller annet forventes det at arbeidet er et resultat av studentens egen innsats. Å utgi andres arbeid for sitt eget er uetisk og kan medføre sterke reaksjoner fra Ifis side. Derfor gjelder følgende:

1. Deling (både i elektronisk- og papirform) eller kopiering av hele eller deler av løsningen utviklet i forbindelse med de obligatoriske og karaktergivende oppgavene i kurset er ikke tillatt.
2. Deling/distribuering av kode og oppgavetekst med personer som ikke er eksamensmeldt i inf1060 dette semesteret, med unntak av kursledelsen og gruppelærere, er ikke tillatt.
3. Hente kode fra annet hold, f.eks. fra andre ``open source'' prosjekter eller kode funnet på nettet er ikke tillatt.
4. Det er greit å få generelle hint om hvorledes en oppgave kan løses, men dette skal eventuelt brukes som grunnlag for egen løsning og ikke kopieres uendret inn.
5. Kursledelsen kan innkalle studenter til samtale om deres innlevering.

Reglene om kopiering betyr ikke at Ifi fraråder samarbeid. Tvert imot, Ifi oppfordrer studentene til å utveksle faglige erfaringer om det meste. Vi oppfordrer til at studentene skal kunne lære av hverandre, men, som sagt, man skal ikke dele/distribuere/kopiere noen form for kode. Det som kreves er som nevnt at man kan stå inne for det som leveres. Hvis du er i tvil om hva som er lovlig samarbeid, kan du kontakte gruppelærere eller faglærer.

Se ellers IFIs [retningslinjer](#)

Lykke til!

Michael, Tor og Pål