

Historikken til dokumentet

2017-03-08: Første versjon publisert kl. 19:00. Endring kan forekomme.

Formål

Blir kjent med grunnleggende filbehandling på flere plattformer vha. Go programmeringsmiljø.

Oppgave 1

Alle filene som det blir referert til her ligger i repository **REP_ICA04**:

<https://github.com/uia-worker/is105-ica04>

a)

Les text1.txt og text2.txt vha. fileutils inn i “byte slices” og sammenlign disse. Filene har tilsynelatende lik innhold, men hvorfor er den ene filen større enn den andre? Bortsett fra forskjeller i størrelsen, hvorfor må man være oppmerksom på dette “fenomenet”?

b)

Skriv en funksjon, som finner ut hvilken kode for linjeskift en tekst-fil bruker. Dere skal selv finne eget navn for deres funksjoner, men programmet skal hete “lineshift.go”. Programmet skal kunne bli utført fra kommandolinje og returnere en eller annen type konklusjon. Programmet skal ta et <filnavn> som in-data.

Oppgave 2

Deler a) og b) utføres i skyinstansen.

Delen c) utføres på MS Windows eller Mac OS X systemer.

Studer “type FileMode” og påfølgende eksemplet i

<https://golang.org/pkg/os/#FileMode>

a)

Skriv et Golang program **fileinfo.go**, som kan utføres fra kommandolinje.

Programmet skal ta et argument, som skal være et filnavn.

Stien til filnavnet kan være enten relativ eller absolutt.

Kommandoet for å utføre programmet skal se slikt ut:

```
fileinfo -f <filnavn>
```

Programmet skal returnere detaljert informasjon om en fil med <filnavn>.

For eksempel, ut-data for kommandoen:

```
fileinfo -f <filnavn>
```

skal være:

```
Information about a file <filnavn_med_full_sti>:  
Size: X in bytes, kibibytes, mibibytes and gibibytes  
Is/Is not a directory  
Is/Is not a regular file  
Has Unix permission bits: -rwxrwxrwx  
Is/Is not append only  
Is/Is not a device file  
Is/Is not a Unix character device  
Is/Is not a Unix block device  
Is/Is not a symbolic link
```

b)

Forklar resultatet, som deres program returnerer for filene “/dev/stdin” og “/dev/ram0”, når det utføres i et Linux-miljø (i skyinstansen).

c)

Kompiler programmet fileinfo.go for MS Windows og/eller Mac OS X operativsystemer.

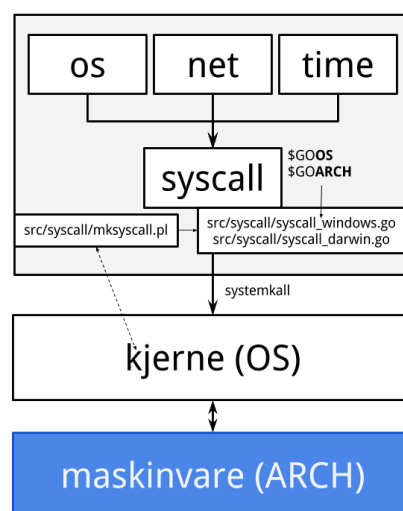
Test programmet på disse platformene og dokumenter eventuelle forskjeller.

Prøv å forklare disse forskjellene basert på informasjon, som dere har søkt opp selv.

Her er informasjon om fil-attributter i MS Windows miljøer:

[https://msdn.microsoft.com/en-us/library/windows/desktop/gg258116\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/gg258116(v=vs.85).aspx)

Figuren som illustrerer arkitekturen for Golang:



Eksempler for kryss-kompilering og kopiering til skyinstansen (alternativt kan man “pushe” til en github-repository fra lokal maskin og gjøre “pull” mot den samme repository fra skyinstansen):

```
GOOS=linux GOARCH=amd64 go build fileinfo.go  
scp osinternals ubuntu@<ip_adresse>:/home/ubuntu/temp
```

```
GOOS=windows GOARCH=amd64 go build fileinfo.go  
GOOS=darwin GOARCH=amd64 go build fileinfo.go
```

Oppgave 3

Studer http://www.devdungeon.com/content/working-files-go#write_bytes

a)

Klassifiser og forklar hvilken type metoder finnes det i Go for å jobbe med filer.

b)

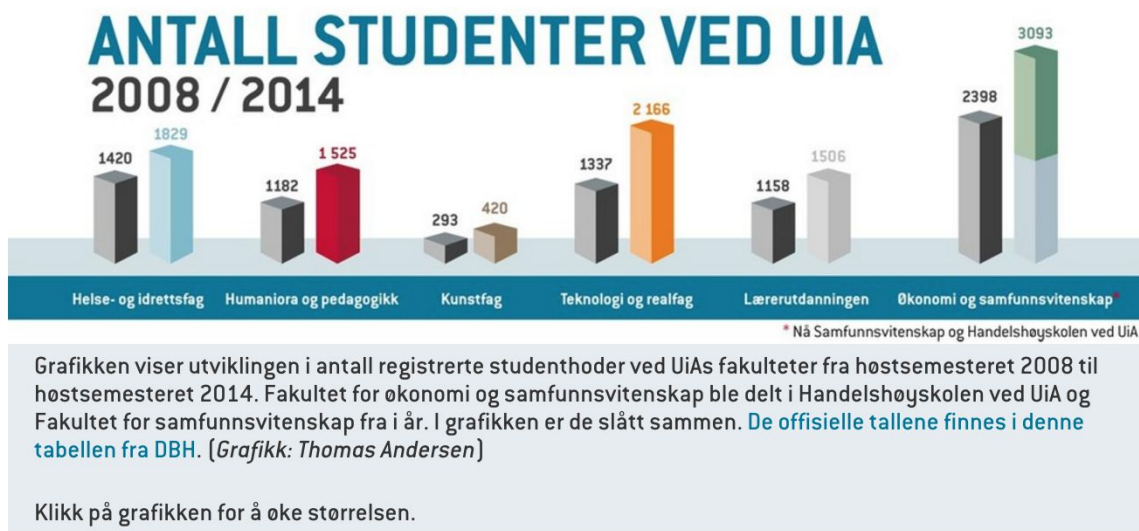
Skriv programmer for alle type metoder for å lese og skrive til filer. Programmene skal lese inn en tekst-fil (flere middels store filer er i repository), skrive ut totalt antall linjer og antall for fem “runes” (bokstaver/tegn), som forekommer mest i disse filene.

c)

Lag benchmarks-tester for alle type metoder for lesing av fil. Forklar fordeler og ulemper for hver av disse metodene i forhold til oppgaven deres.

Dere kan ha behov til å se mer på slices i Go: <https://blog.golang.org/slices>

Oppgave 4



(Fakultet for helse- og idrettsfag har skiftet navn, og heter i dag Fakultet for helse- og idrettsvitenskap.)

Publisert: 17. oktober 2014

a)

Basert på illustrasjon over, lag en tabell med kolloner "UiAs fakultet", "Antall studenter" og "Sannsynlighet" (for at en UiA student hører til det spesifikke fakultetet) for høstsemester 2014.

b)

Når du lærer (får informasjon) om at en tilfeldig valgt student hører til et spesifikt fakultet, for hvilket fakultet får du MINST informasjon?

c)

Design en variabel-størrelse Huffman kode, som minimaliserer gjennomsnittsantall av bits i en melding som er kodet for fakulteter og tilfeldig valgte grupper av studenter. Vis et binært tre for Huffman kode og gi kode og kodelengde for hver kurs. Ble gjennomgått på forelesning 2017-03-08.

Referanser:

<https://users.cs.cf.ac.uk/Dave.Marshall/Multimedia/node210.html>

<https://www.youtube.com/watch?v=apcCVfXfcqE>

d)

Hva er gjennomsnittslengden for en melding som inneholder fakultetskoder for 100 tilfeldig valgte studenter? Beregn og forklar.

e)

Lag en algoritme som kan kode og dekode med den valgte koden. Algoritme skal først forklares og så implementeres i Golang. Dere velger selv navn og struktur for deres funksjoner. Implementer tester med forskjellige type inn-data (korrekt kodet og feil kodet).