

Forkurs i informatikk - Tirsdag

Dagens formål: Få praktisk trening og bli kjent med det kommandobaserte programmet terminalen.

Motivasjon: Når man programmerer eller raskt skal manøvrere seg gjennom mapper og filer, er det nyttig for oss å kunne bruke kommandoer for å få datamaskinen til å utføre det vi ønsker. Når man har blitt vant med å bruke terminalen vil man kunne arbeide raskere.

Hjelp: dersom det er noen ord eller uttrykk du ikke forstår står det en del forklart [her](#). Mye brukte kommandoer ligger [her](#).

Ikke nøl med å spørre om hjelp underveis!

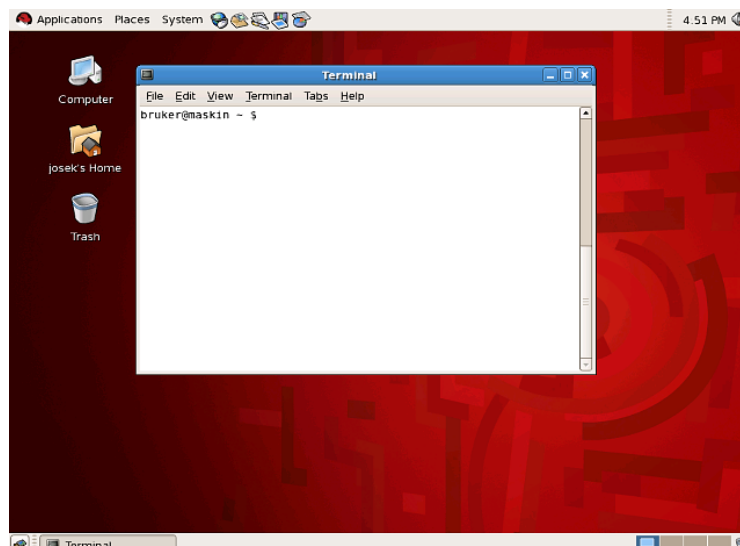
Om programmet vi skal bruke

En Terminal er et kommandobasert program, som finnes for Linux og Mac-baserte maskiner. Man finner også noe lignende på Windows - kalt kommandolinjevindu (CMD). Dette betyr at vi bruker den til å fortelle datamaskinen hva den skal gjøre ved hjelp av instruksjoner. Instruksjonene vi gir kommer i form av *kommandoer*, enkle setninger vi skriver til Terminalen. Dette skal vi se nærmere på straks. Kort oppsummert brukes Terminalen til:

- Å ta i mot kommandoer og behandle de
- Å sende data inn til en datamaskin eller et system
- Å presentere data fra en lokal eller ekstern datamaskin

Terminalen - et første møte

Nå skal vi ta en første titt på Terminalen. Om du har en Windows-maskin foran deg - klikk på Linux-ikonet på skrivebordet for å logge inn. Når man er pålogget en Linux-maskin på IFI, åpnes det automatisk et terminalvindu. Terminalen er det sorte vinduet øverst i venstre hjørne. I denne boksen kan vi skrive kommandoer for å fortelle datamaskinen hva den skal gjøre. Vi bruker med andre ord kommandoer for å få ting gjort raskere enn med musepeker og vinduer. Terminalvinduet ser slik ut:



Om du ikke har et slikt vindu åpent på din maskin, kan man høyreklikke på skrivebordet og velge "Open in Terminal". Ledeteksten, prompt på engelsk, er en liten og rar tekst som vises helt til venstre i terminalvinduet. Denne ser typisk slik ut:

```
[bruker@maskin ~] $
```

Her angis brukernavnet, hvilken maskin man er logget inn på, og hvor du befinner deg i filsystemet: "~" angir her at du befinner deg på roten av hjemmeområdet ditt. Med hjemmeområdet ditt menes din private plass hos IFI - hvor du har typiske mapper som "Documents" og "Downloads". Hjemmeområdet er med andre ord en mappe hvor filene til brukeren lagres. Dette hjemmeområdet vil alltid ha samme navn som ditt brukernavn ved UiO. I vinduet vil kommandoer blir skrevet inn etter dollartegnet.

Forsøk nå å skriv kommandoen **ls**, etterfulgt av enter-tasten for å kjøre den. **ls** står for **list files** - og er kommandoen for å få en oversikt over alle filer og mapper der hvor man befinner seg.

Filene og mappene som ligger på ditt hjemmeområde vil nå bli listet opp i terminalen. For å se hvordan de samme filene ser ut i et vanlig vindu, åpne mappen med hjem-ikonet på skrivebordet (det er mulig denne ligger bak terminal-vinduet). Vi viser med andre ord frem filene ved hjelp av tekst i stedet for bilder.

Gå nå tilbake i Terminalen og kjør kommandoen **touch**, som vist under:

```
[bruker@maskin ~] $ touch mittnavn.txt
```

Se nå om du kan finne igjen tekstfilen du nettopp opprettet i hjemmeområdet-mappen. Du har nå laget en ny fil, kun ved hjelp av en kommando!

Dette illustrerer det som ble snakket om på forelesning i dag - forskjellen mellom den vindusbaserte måten å jobbe på, og den kommandobaserte måten å jobbe på. Når vi venner oss til å bruke Terminalen som et sted å arbeide i, kan vi ofte jobbe raskere.

Vindusbasert

Pek og klikk
Visuelle fremstillinger

Kommandobasert

Skrive inn kommandoer
Tekstlig fremstilling

Når vi jobber i terminalen kan det være nyttig for oss å se hvor i mappestrukturen vi befinner oss. For dette formålet finnes det selvfølgelig en egen kommando, **pwd** (**print working directory**), som forteller oss nettopp hvor vi befinner oss. Kjør denne kommandoen. Forhåpentligvis får dere nå opp noe lignende:

```
[bruker@maskin ~] $ pwd  
/uio/hume/student-u62/bruker
```

Vi ønsker nå at alle skal være på hjemmeområdet sitt. Hvis brukernavnet/gjestebbrukernavnet vises til slutt i resultatet fra **pwd**, er du allerede der (som vist på sist bilde). Hvis dette ikke er tilfelle, kan man kjøre kommandoen **cd** for å komme tilbake til hjemmeområdet:

```
[bruker@maskin ~] $ cd
```

Hvis man kjører kommandoen **cd** uten noe annet, vil den ta deg til hjemmeområdet ditt. Den har likevel også andre bruksområder, som vi skal se om litt.

Forsikre deg nå om at du befinner deg på hjemmeområdet ved å kjøre kommandoen **pwd**.

Laste ned forkursmappen

Vi skal nå jobbe videre med en zip-mappe med filer. Denne mappen skal vi laste ned til hjemmeområdet, og pakke ut. For å gjøre dette skal vi aller først lage en ny mappe - kalt "*mineFiler*". Når vi legger alt som har med forkurset å gjøre inn i en mappe, blir det lettere for oss å holde oversikt.

Skriv nå kommandoen **mkdir** til Terminalen, **make directory**, og send med "*mineFiler*" som *argument*. At vi sender noe som argument betyr at vi skriver noe etter selve kommandoen, som *mineFiler* under:

```
[bruker@maskin ~] $ mkdir mineFiler
```

*"Vi sender **mineFiler** som argument til **mkdir**"*

Vi har nå laget en ny mappe! Kjør **ls** for å se at den nyopprettede mappen nå ligger i listen. Stemmer det?

Flere bruksområder - cd

Nå skal vi bruke kommandoen **cd** (**change directory**) på en litt annen måte enn vi gjorde i sted (husker du at vi kunne flytte oss tilbake til hjemmeområdet med den?). Vi skal nå *sende med et argument* til **cd** for å flytte oss *inn* i mappen vi nettopp lagde. Kjør nå kommandoen under:

```
[bruker@maskin ~/] $ cd MineFiler
```

"Flytt meg fra der hvor jeg er (cd) til mappen som heter mineFiler"

Her er **cd** kommandoen og *mineFiler* *argumentet*. Kjør kommandoen **pwd** for å se at du nå befinner deg inne i mappen kalt *mineFiler*.

Nå som vi er inne skal vi starte med å laste ned et zip-arkiv (mappe) fra nettsiden til forkurset - dette selvfølgelig med en egen kommando. Til dette skal vi nemlig bruke kommandoen **wget** (**WWW get**), som laster ned innhold fra en nettside - uten bruk av en nettleser(!).

Kjør nå kommandoen **wget** og send linken under som argument:

<http://www.uio.no/studier/emner/matnat/ifi/INF0010/h15/forkurs.zip>

Når **wget** kjøres vil den laste ned filen (i dette tilfellet en zip-fil) til den mappen man befinner seg i. Hvis du befant deg i forkurs-mappen når kommandoen ble kjørt, vil innholdet nå altså ligge der. Dette gjelder for de aller fleste kommandoer i Terminalen - kommandoen får konsekvenser for den mappen du befinner deg i.

For å sjekke at zip-filen ble lastet ned, kjør kommandoen **ls** og se at du finner filen "*forkurs.zip*". For å pakke opp den zippede mappen kjører du kommandoen **unzip** med "*forkurs.zip*" som argument:

```
[bruker@maskin ~/MineFiler] $ unzip forkurs.zip
```

Fra **unzip**-kommandoen får også vi beskjed/output om hvilke filer som blir pakket ut fra forkurs.zip. Skriv nå **ls** på nytt for å se at du har fått filene pakket ut.

Oversikt og litt opprydning

Nå har vi skrevet inn en del kommandoer i Terminalen. Hvis man ønsker å se på alle de siste kommandoene som har blitt skrevet inn, finnes det selvfølgelig en kommando for det - **history**.

Nå fikk du kanskje mye utskrift til Terminalen? Hvis du ønsker å fjerne all teksten, kan du kjøre kommandoen **clear** som gjør nettopp dette. Denne kan være spesielt nyttig når vi senere skal programmere, for å få oversikt over tilbakemeldinger fra programmene våre. Prøv dette nå.

Nå som vi har pakket ut zip-mappen vår, skal vi se nærmere på hva den inneholder. Kjør kommandoen **tree**, som fremstiller alle filene og mappene i forkursmappen på en oversiktlig måte (denne ligner **ls**-kommandoen). **tree** burde gi følgende utskrift:

```
[bruker@maskin ~/MineFiler] $ tree
.
|-- Java
|   |-- Dag2.java
|   |-- Morro.java
|   |-- ProgOppgave2.java
|   |-- ProgOppgave3.java
|   |-- Skjelett.java
|-- Python
|   |-- Dag2.py
|   |-- Morro.py
|   |-- ProgOppgave2.py
|   |-- ProgOppgave3.py
|-- GoT.txt
|-- LOTR.txt
|-- treSmaaKinesere.txt
```

Kjør gjerne også **ls** for å se forskjeller/likheter mellom disse to.

Kommandoer for å jobbe med filer

Naviger nå inn i mappen “*forkurs*”, ved å bruke kommandoen **cd** som tidligere. Nå skal vi kjøre kommandoen **file** på en av filene inne i forkurs-mappen. Denne kommandoen viser informasjon om filer som kan være nyttig for oss ved enkelte tilfeller. Prøv eksempelvis:

```
[bruker@maskin ~/MineFiler] $ file LOTR.txt
```

Hva blir skrevet ut? Prøv nå å kjør kommandoen under:

```
[bruker@maskin ~/MineFiler] $ file Java
```

Her forteller **file**-kommandoen oss at vi har å gjøre med en mappe. Dette vil blant annet si at vi kan bruke **cd** for å navigere oss inn i den.

Nå skal vi åpne en av filene i forkurs-mappen for å se på innholdet. Til dette kan vi bruke kommandoen **cat** - som dumper alt innholdet i en fil som ren tekst til Terminal-vinduet. Kjør kommandoen under:

```
[bruker@maskin ~/MineFiler] $ cat GoT.txt
```

Som du ser nå blir det upraktisk å få veldig mye tekst på en gang. Til dette finnes det selvfølgelig en egen kommando - **less** - som aldri viser mer enn ett skjermbilde av gangen. Prøv nå kommandoen **less** på samme fil. Nå kan man trykke enter for å lese videre i tekstfilen.

Prøv gjerne å åpne “*GoT.txt*” på vanlig måte, altså uten Terminalen ved å dobbeltklikke på filen i utforskeren.

Nå skal vi prøve å kopiere filen “*treeSmaaKinesere.txt*”. Dette gjør vi med kommandoen **cp** (**copy**). For første gang tar kommandoen nå to argumenter;

- Første argument - Navnet på filen vi skal kopiere
- Andre argument - Navnet på kopien vi skal lage

Lag en kopi ved å kjøre **cp** som vist under:

```
[bruker@maskin ~/MineFiler] $ cp treeSmaaKinesere.txt treeSmaa.kopi
```

Som du ser kan vi gi nye filer den filendelsen vi selv ønsker, her “*.kopi*”. Sjekk nå at du har fått en kopi av filen, ved hjelp av kommandoen **ls**.

Terminalen - tips og triks

- For å unngå å bruke for lang tid på å skrive inn kommandoer, kan man også bruke piltast opp eller piltast ned for å raskt kunne velge tidligere kommandoer man har skrevet inn.
- I de fleste Unix/Linux-systemer har store og små bokstaver stor betydning. Det vil si at `ls` og `Ls` betyr to ulike ting. På samme måte vil en fil med navn "*ifi.txt*" og en annen med "*lfi.txt*" angi helt ulike filer. Dette gjelder både når vi bruker Terminalen, i Linux og i programmering generelt.
- Med Tab-tasten kan du automatisk fullføre kommando- eller filnavn i terminalen. Om det finnes flere muligheter, altså to kommandoer eller filer begynner på de bokstavene du har skrevet inn, kan du trykke Tab-tasten to ganger for å få vist alle alternativene. Om du skriver "`le`" og trykker Tab-tasten to ganger får du opp en liste over kommandoer som begynner på "`le`". Skriver du en ekstra "`s`" og trykker Tab-tasten igjen vil det dukke opp "`less`" som er det korteste felles treffet for kommandoer som begynner på "`les`". Trykker du Tab-tasten igjen vil du få se andre kommandoer enn `less` som begynner på "`less`".
- Om du ønsker å kopiere tekst i Linux holder det å markere teksten. Tekst kan deretter limes inn ved å klikke inn rullehjulet på musen. Dette kan være spesielt nyttig for å lime inn tekst i terminalen.

Sletting og flytting

Når vi programmerer får vi av og til overflødige filer som vi ønsker å slette. Sletting av filer kan også gjøres i Terminalen - med kommandoen **rm** (**r**emove). I mappen vår ligger det en fil som nå skal slettes, nemlig filen kalt "*temp.txt*". Slett filen ved å sende med filnavnet som argument til

Vær forsiktig med kommandoen **rm** - den gir den ingen advarsler ved sletting!

kommandoen **rm**. Når du har gjort det, kjøp **ls** for å se at den nå har blitt slettet. Slett nå også "*treeSmaa.kopi*" på samme måte. Forkursmappen vår er fortsatt litt rotete, med filer og mapper om hverandre. Lag derfor en ny mappe med **mkdir**-kommandoen som du kaller for "*Tekstfiler*".

Inne i denne mappen skal vi nå legge alle filer som ender på "*.txt*". Bruk nå det du har lært fra de tidligere øvelsene til å flytte filene ved hjelp av kommandoen **mv** (**m**ove). Denne tar også to argumenter, og fungerer på følgende måte:

mv

- Første argument - Navn på filen som skal flyttes
- Andre argument - Navn på mappen den skal flyttes til

Hint til å sjekke at du har gjort øvelsen riktig: **mv**, **cd**, **ls**.

Starte programmer

Det siste vi skal gjøre i dag er å se på to ulike tekstredigeringsprogrammer fra Terminalen, som vi kommer til å få bruk for neste gang. Slike programmer, kalt editorer, bruker vi når vi skal programmere. Det er opp til dere selv hvilken dere ønsker å bruke til å programmere med, og vi ønsker derfor nå at dere skal teste to av de.

I Terminalen trenger vi kun å skrive inn navnet på editoren vi ønsker å åpne, med et &-tegn som argument:

```
[bruker@maskin ~/MineFiler] $ emacs &
```

Med kommandoen over vil man starte opp editoren Emacs i et eget vindu. Prøv nå å skriv en tekstfil og lagre den i forkursmappen.

Når vi sender med &-tegnet som argument, forteller vi Terminalen at vi fortsatt ønsker å bruke den selvom programmet vi åpnet er oppe.

Når du har prøvd ut Emacs, kan du prøve editoren Atom:

```
[bruker@maskin ~/MineFiler] $ atom &
```

Prøv også her å skriv en tekstfil og lagre den i forkursmappen.

Oppgaver fullført

Bra jobbet - du er nå ferdig med dagens øvelser! Bruk gjerne tid på repetisjon eller å bli kjent med en av editorene - disse skal vi jobbe mer med neste gang! Husk at du ikke trenger å pugge kommandoene vi har vært borti i dag, og at dette mest er ment som øvelser som skal hjelpe deg på vei. Det viktigste du kan huske på fra dagens øvelser er:

- Kommandoen **ls** som viser filer i en mappe
- Kommandoen **cd** som skifter mellom mapper
- Kommandoen **pwd** for å se hvor du er hen i mappestrukturen

Synes du dette var gøy? Ta gjerne kontakt med et orakel på rommet ditt for litt mer avanserte øvelser.

Psst! Lurer du på hvilke andre editorer som finnes der ute? Kikk på Sublime Text 2, Notepad++, Gedit, eller Vim.