

Introduktion til terminalen

Lars Nielsen

20. april 2014

Indhold

1	Introduktion	5
1.1	Kommandoer	5
2	Filsystem	7
2.1	PWD	7
2.2	CD	7
2.3	LS	8
2.4	MKDIR	8
2.5	RMDIR	9
2.6	RM	10

Introduktion

I dette kapitel introduceres principperne omkring Terminal og hvordan den benyttes.

Terminalen er som før nævn en applikation, som giver en standard kommandolinje miljø. Hvis man installere en linux distribution som ArchLinux, får man adgang til et ren CLI-brugergrænseflade. Der imod installer man en linux distribution som Fedora som kommer med en grafiskbrugergrænseflade. Har man adgang til et program som emulere terminale, dette program kan havde forskellige navne (eks. XTerm, Terminal eller Konsole). Begge typer kaldes terminal og som giver adgang til en masse applikationer og system programmer/procedure.

1.1 Kommandoer

For at kunne åbne programmer eller procedure i terminale, skal man kende den korrekt kommando. Det vil sige det "navn" som er associeret med en program eller en procedure. Eksempler på kommandoer kunne være:

- ls
- mkdir
- df

Kommando struktur

Alle kommandoer har en fælles struktur, der er fælles for alle kommandoer. Denne struktur er derfor virkelig vigtigt at få slået fast og have kendskab til. Denne struktur er således opbygget; kommando flag argumenter.

Kommando er som førnævnt navnet på det program eller den procedure man ønsker at kalde.

Flag er *indstillinger* som fortæller et program eller en procedure hvordan en opgave skal udføres. En kommando kan tage op til flere flag.

Argumenter er det/de elementer som en kommando skal udføres på.

```
$: ls -a ~/Desktop
```

kommando eksempel

Filsystem

En del af det at benytte en computer, er at kunne navigere filsystemet hurtigt og effektivt. Derfor er det vigtigt at kunne benytte de korrekte kommandoer, til disse arbejdes processer. Disse kommandoer vil blive dækket i dette kapitel.

2.1 PWD

Print Working Directory printer den sti, som man befinder sig i. Kommandoen tillader således en bruger at orientere sig om, hvor i filsystemet denne befinder sig.

```
$:version_3 tools$ pwd  
/Users/tools/Documents/intro_til_terminal/version_3
```

pwd eksempel

2.2 CD

Change Directory ændre den sti, man befinder sig. Kommandoen tager et parameter, som skal være en sti eller `..` som føre en bruger en niveau op i filsystemet.

```
$:version_3 tools$ pwd  
/Users/tools/Documents/intro_til_terminal/version_3  
$:version_3 tools$ cd code/  
$:code tools$ pwd
```

```
/Users/tools/Documents/intro_til_terminal/version_3/code
```

cd eksempel

2.3 LS

List directory content *S* lister, som standard alle *ikke skjulte* filer.

```
$:version_3 tools$ ls
code          filesystem    main_book.log
compile.sh    introduktion  main_book.pdf
compile.sh~   main_book.aux  main_book.tex
```

ls eksempel

Hvis en bruger ønsker at få vist skjulte filer, tilføjer man flaget *-a*, som betyder at man får vist alt

```
$:version_3 tools$ ls
.      code          filesystem    main_book.log
..     compile.sh    introduktion  main_book.pdf
```

ls -a eksempel

2.4 MKDIR

*MaKe DI*Rectory tillader en bruger at lave en mappe. Kommandoen tager et argument, som er navnet på den nye mappe.

```
$:demo tools$ ls
$:demo tools$ mkdir dir_ex
$:demo tools$ ls
dir_ex
```

mkdir eksempel

Hvis man ønsker at lave en mappe et andet sted i systemet, kan man give en fuld sti som parameter til kommandoen


```
$:Desktop tools$ pwd
/Users/tools/Desktop
$:Desktop tools$ mkdir /Users/tools/Documents/
    intro_til_terminal/version_3/demo/dir_2_ex/
$:Desktop tools$ ls /Users/tools/Documents/
    intro_til_terminal/version_3/demo/
dir_2_ex          dir_ex
```

mkdir med sti eksempel

Dir 1: */Users/tools/Documents/intro_til_terminal/version_3/demo/dir_2_ex/*

Dir 2: */Users/tools/Documents/intro_til_terminal/version_3/demo/*

Det er også muligt at oprette flere mapper på engang, ved at skrive navne på mappe, med et mellemrum

```
$:demo tools$ ls
dir_ex
$:demo tools$ mkdir dir_ex_2 dir_ex_3
$:demo tools$ ls
dir_ex          dir_ex_2          dir_ex_3
```

mkdir med flere mapper eksempel

2.5 RMDIR

ReMove DIRectory tillader at en bruger at fjerne en tom mappe. Kommandoen tager et argument, som er navnet på den nye mappe og kan som *mkdir* også tage stien til en mappe.

```
$:demo tools$ ls
dir_2_ex          dir_ex
$:demo tools$ rmdir dir_2_ex/
$:demo tools$ ls
dir_ex
```

rmdir eksempel

2.6 RM

ReMove tillader en at fjern en fil fra en mappe og tager pågældende fil som argument

```
$:demo tools$ ls
demo.txt      dir_ex
$:demo tools$ rm dir_ex/
rm: dir_ex/: is a directory
$:demo tools$ rm demo.txt
$:demo tools$ ls
dir_ex
```

rmdir eksempel

Man kan også fjerne flere filer og under mapper og en hoved mappe på en gang. Dette gøres ved at bruge flaget *-r*, hvor *r* betyder rekursivt.

```
$:demo tools$ ls
demo_2  dir_ex
$:demo tools$ ls demo_2/
demo.txt
$:demo tools$ rm demo_2/
rm: demo_2/: is a directory
$:demo tools$ rm -r demo_2/
$:demo tools$ ls
dir_ex
```

rmdir -r eksempel

2.7 MV