

Introduktion til Terminal

Lars Nielsen
lnc13.lars@gmail.com

10. december 2013

Forord

Denne bog er tiltænkt nybegynder (n00bz) inden for *NIX¹ verden. Som godt kunne tænke sig at få en introduktion i hvad der ligger under den poleret brugergrænseflade.

Om forfatteren

Lars er født i 1989 så et lævn fra det forige årtusind. Lars er uddannet datamatiker fra Aarhus Erhvervs Akademi sommeren 2013. Men han start med at læse datalogi ved Aarhus Universitet i 2009, men det var nu lidt for teoretisk, så han skiftet til den mere praktisk tilgang og læser nu software ved Aalborg Universtet under School of information and computer technology. Desuden har Lars arbejdet ved flere virksomheder der udarbejder webbaseret løsninger og nogle få virksomheder der arbejder primært med back-end løsninger. Kort sagt Lars er nørd, og bruger det mest af sin tid med softwareudvikling og primært ligger hans interesse inden for back-end, system udvikling og systemhåndtering.

Kontakt:

Hvis du har kommentar, forslag eller andre ting til denne bog, ville det være at fortrække at smide en kommentar på bogens github side, som findes her github.com/loopTools/intro_til_terminal. Hvor der findes en issue side, hvor sådanne ting bliver håndteret. Har du derimod et spørgsmål til Lars selv kan du kontakt ham på følgende mail adresse: lnc13.lars@gmail.com

Tak til:

Tak til www.linux.dk og holdet bag, for at give mig lov til at bruge siden til at start min serie af guides, som er blevet samlet til den her bog.

¹*NIX er UNIX eller Linux baseret systemer

Tak til Peter Lyberth og Kim Rostgaard Christensen, som begge står bag linux.dk og som af og til har læst nogle af online guidesne igennem, for stavefejl.

Tak til folkne bag L^AT_EX, som bogen er opbygget i.

Indhold

1	Introduktion	1
1.1	Hvad bliver ikke dækket	1
1.2	Kommando opbygning	2
1.3	MAN - Din ven i mørket	2
2	Navigering i filsystemet	5
2.1	PWD - hvor er vi?	5
2.2	LS - Hvad er der her?	6
2.3	CD - Jeg vil væk	7
2.4	MKDIR - Ny mappe	7
2.5	CP - Kopi	7
2.6	MV - Flyt dig	8
2.7	RM - Væk med dig	8
A	Ordliste	11

Kapitel 1

Introduktion

Siden den personlige computeres oprindelse, har der været to primære input enheder til computer den ene er pegeværktøjet (musen, pointwheel trackball) og tastaturet. De fleste bruger i dag en kombination af de to, til at navigere og bruge deres computere. Så er der folk som os (nørder) som fortrækker at flytte hænderne mindst muligt fra tastaturet, medmindre vi rækker hånden ud af kaffen. Vi elsker at kunne bruge vores computere og systemer uden at skulle klikke rundt. Vi har et hemmeligt våben til dette, dette våben er bedst kendt som Terminal eller kommandolinje.

Dette våben er et lævn fra den gang computer ikke havde en mus og kun tastaturet som input enhed.

Terminal eller Terminalen er et kommandolinje værktøj som findes i forskellig udgaver, men de fleste systemer bruger de samme kommandoer. De kommandoer som bliver brugt i denne bog kan bruges på Mac OS X, Linux og BSD, hvis en funktion ikke findes på alle systemer bliver dette naturligvis noteret.

1.1 Hvad bliver ikke dækket

Der er elementer af den kommando linje baseret verden, som ikke bliver dækket i denne bog og med god grund.

Server opsætning er et helt emne for sig selv, og er simpelt hen for stort til at blive dækket i en begynder bog på dette niveau og det varierer meget fra system til system.

Tekstværktøjer som en del af det at bruge kommandolinjen, er tekstværktøjer og der findes forskellig slags (Vi(m), Emacs, Nano, Pico, ...). Og da der er en kæmpe religionskrig i mellem specielt Vi(m) og Emacs har jeg valgt

ikke at dække dem. I bogen bruges Vi eller Emacs. Jeg vil dog prøve at holde mig til Vi, da flest systemer kommer med denne som standard.

Hacking er et mere avanceret emne og kræver lidt evner og bliver derfor ikke dækket i denne bog.

1.2 Kommando opbygning

For at kunne benytte kommandolinjen eller terminalen, benyttes kommandoer. Disse kommandoer har en opbygning og for at benytte disse kommandoer er det ret vigtigt at forstå denne opbygning. En Kommando består primært af tre dele, disse tre dele kaldes kommandoen, flag og argumenter.

Kommandoen

Er navnet på den kommando som man ønsker at benytte sig af. Navnet er altid den først del af et kommandokald. Eksempler på kommandonavne er; *cd*, *ls*, *pwd* og *mv*.

Flag

Flag er parameter, som benyttes til at ændre eller manipulere det output en kommando returnerer. Disse kommer ofte efter kommandonavnet eksempelvis; *ls -a* som bliver beskrevet senere. Flag starter ofte med et dash (-) og derefter et enkelt bogstav. En kommando kan sagtens tage flere flag eller ingen, det ser vi eksempler på senere.

Argumenter

Disse kommer ofte efter flag og er det "element" som en kommando bliver udført, forskellig kommando kan tage mere end et argument. Eksempel på kommando uden flag, men med et argument *ls /test_2*

1.3 MAN - Din ven i mørket

Når man vil kende sin funktion lidt mere i dybden, bruger man en kommando som hedder *man*. *man* står for manual og giver et overblik over kommando, mulige flag og argumenter. se eksempel neden for

```
h271: intro_til_terminal tools$ man ls
LS(1)                      BSD General Commands Manual
                           LS(1)
```

NAME

ls -- list directory contents

SYNOPSIS

ls [-ABCFGHLOPRSTUW@abcdefghijklmnopqrstuvwxyz1] [file ...]

DESCRIPTION

For each operand that names a file of a type other than directory, `ls` displays its name as well as any requested, associated information. For each operand that names a file of type directory, `ls` displays the names of files contained within that directory, as well as any requested, associated information.

If no operands are given, the contents of the current directory are displayed. If more than one operand is given, non-directory operands are displayed first; directory and non-directory operands are sorted separately and in lexicographical order.

The following options are available:

man siden kan navigeres med med piletasterne og man kommer udaf den ved at indtaste *q*. Hvis du vil vide lidt mere om *man*, så har *man* selv en *man* side og den tilgås ved at skrive *man man*

Kapitel 2

Navigering i filsystemet

En af de vigtigste ting at kunne i kommandolinjen, er at have evnerne til at navigere OS'ets filsystem. Derfor starter vi med at forklare de forskellige kommandoer til at finde rundt.

2.1 PWD - hvor er vi?

En ret relevant del af navigering er ofte at vide hvor man i det hele taget er. Det er her kommandoen *pwd* kommer ind i billedet. *pwd* står for **P**rint **W**orking **D**irectory, kommandoen returnerer den sti som man arbejder i lige nu.

```
h271: intro_til_terminal tools $ pwd
/Users/tools/Documents/intro_til_terminal
```

Og hvad så

Hvad kan man så bruge det til? Man kan altid bruge ens nuværende "position" til at navigere filsystemet. Men for at kunne bruge information skal man kunne forstå resultat, så lad os bryde stien ned.

intro_til_terminal

Det sidste element af stien er den mappe, man arbejder i og interagerer med. Det er altså her andre kommandoer som interagerer med filer og visse andre kommandoer udføre deres arbejde hvis de bliver kaldt.

/Users/tools/Documents/

Er den overordnet filsti, som man er nød til at komme igennem for at kunne komme til *intro_til_terminal*.

2.2 LS - Hvad er der her?

En anden vigtig del af at arbejde i filsystemet, er at vide hvilke filer og mapper, som er i den mappe man arbejder i. Det er til dette formål man bruger kommandoen *ls*. *ls* betyder list directory contents, altså list indholdet af en mappe. Hvis kommandoen kaldes ud flag, returneres en liste af alt, som ikke er skjult i mappen.

```
h271: intro_til_terminal tools$ ls
LICENSE      fil_sys_nav .tex intro.tex      main_book.tex
README.md    forord.aux   main_book.aux main_book.toc
compile.sh    forord.tex   main_book.log  ordliste.aux
fil_sys_nav .aux intro.aux    main_book.pdf  ordliste.tex
```

Men jeg har skjulte filer

På UNIX baseret, systemer er alle filer og mapper som starter med *.* skjulte, eksemplet kunne hede *.skjult*. Den vil vi altså gerne kunne se, så vi bruger flaget *-a* som står for all eller alle, også får man en lidt anderledes liste returneret.

```
h271: intro_til_terminal tools$ ls -a
.      LICENSE      fil_sys_nav .tex intro.tex
..     README.md    forord.aux   main_book.aux
.git    compile.sh    forord.tex   main_book.log
.skjult fil_sys_nav .aux intro.aux    main_book.pdf
```

List alt i en undermappe

Man kan også liste alt i mappe ved at give stien til mappe

```
h271:test tools$ ls
t_1  t_2  test_2
h271:test tools$ ls t_1
demo.txt
```

2.3 CD - Jeg vil væk

For at kunne navigere rundt i sit filsystem skal man naturligvis også kunne komme væk fra den mappe man er i. Dette gøres via kommandoen *cd*, som står **C**hange **D**irectory eller ændre mappe.

```
h271: intro_til_terminal tools$ pwd
/Users/tools/Documents/intro_til_terminal
h271: intro_til_terminal tools$ cd test/
h271:test tools$ pwd
/Users/tools/Documents/intro_til_terminal/test
```

Hvis man vil et "trin" op i filsystemet skriver man *cd ..*

2.4 MKDIR - Ny mappe

Man kan få brug for oprette en mappe, dette gøres med kommandoen; *mkdir* som betyder **MaKe Directory** eller lav mappe. Kommandoen opretter en mappe i den nuværende arbejdesmappe.

```
h271:test tools$ pwd
/Users/tools/Documents/intro_til_terminal/test
h271:test tools$ ls
h271:test tools$ mkdir test_2
h271:test tools$ ls
test_2
```

2.5 CP - Kopi

Hvis man ønsker at kopiere en fil, gøres dette med kommandoen *cp* som står for **C**opy. Kommandoen kræver 2 argumenter; filen som ønskes kopieret og destinations mappen. Så hvis man har en mappe *t_1* som indholder en fil *demo.txt* som man ønsker at flyttet til mappen *t_2* gøres det sådan her:

```
h271:test tools$ ls t_1
demo.txt
h271:test tools$ ls t_2
h271:test tools$ cp t_1/demo.txt t_2
h271:test tools$ ls t_1
demo.txt
h271:test tools$ ls t_2
```

```
demo.txt
```

2.6 MV - Flyt dig

MoVe eller flyt er en måde hvor på man kan flytte en fil eller en mappe fra et sted i systemet til et andet. Lige som *cp* tager *mv* to argumenter, elementer som man vil have flyttet og destinations mappen.

```
h271:test tools$ ls t_1
demo.txt
h271:test tools$ ls t_2
h271:test tools$ mv t_1/demo.txt t_2/
h271:test tools$ ls t_1
h271:test tools$ ls t_2
demo.txt
```

Omdøb

Man kan også omdøbe en fil med kommandoen *mv*, dette gøres ved at tage en fil som først parameter og bruger en fil som anden parameter, eksempelvis

```
h271:t_2 tools$ ls
demo.txt
h271:t_2 tools$ mv demo.txt demo_2.txt
:t_2 tools$ ls
demo_2.txt
```

2.7 RM - Væk med dig

ReMove eller fjern er en kommando som kan bruges til at slette filer.

```
h271:test tools$ ls t_2/
demo_2.txt
h271:test tools$ rm t_2/demo_2.txt
h271:test tools$ ls t_2/
```

-r fjør alt under mig og mig

Hvis man ønsker at slette en mappe og alt der ligger i mappen, skal man sætte følgende flag på *rm* -r, hvor -r står for rekrusiv. Hvilket vil sige at den udføre kommandoen rekrusivt.

```
h271:test tools$ ls
t_1      t_2      test_2
h271:test tools$ ls test_2/
demo.2.txt
h271:test tools$ rm -r test_2/
h271:test tools$ ls
t_1      t_2
```

RMDIR - Fjern mappe med indhold

Hvis man vil fjerne en mappe, som man ved er tom, kan man skrive *rmdir* og så mappens navn som argument.

```
h271:test tools$ ls
t_1      t_2
h271:test tools$ ls t_2
h271:test tools$ rmdir t_2/
h271:test tools$ ls
t_1
```

Bilag A

Ordliste

Da ikke alle benytter de samme ord for alt, er der her en ordlist, som forklare ordne som bliver brugt.

Mappe et element som kan indeholde andre mapper og filer. Kaldes også for folder.

***NIX** er et gennerelt udtryk for systemer som basere sig på UNIX og GNU/Linux

***BSD** er en familie af UNIX baseret systemer som bygger på BSD eller Berkeley Software Distrobution, familiemedelmmer er blandt andre Mac OS X, FreeBSD og OpenBSD.

OS X og Mac OS X er det system som Apple Inc. benytter på deres bærebare og stationære computer. OS X er delvist baseret på NextSTEP og FreeBSD.

Parameter disse kaldes også argumenter, men i denne bog brugers argumenter om noget andet.

Dash (-) kaldes normalt bindestreg.