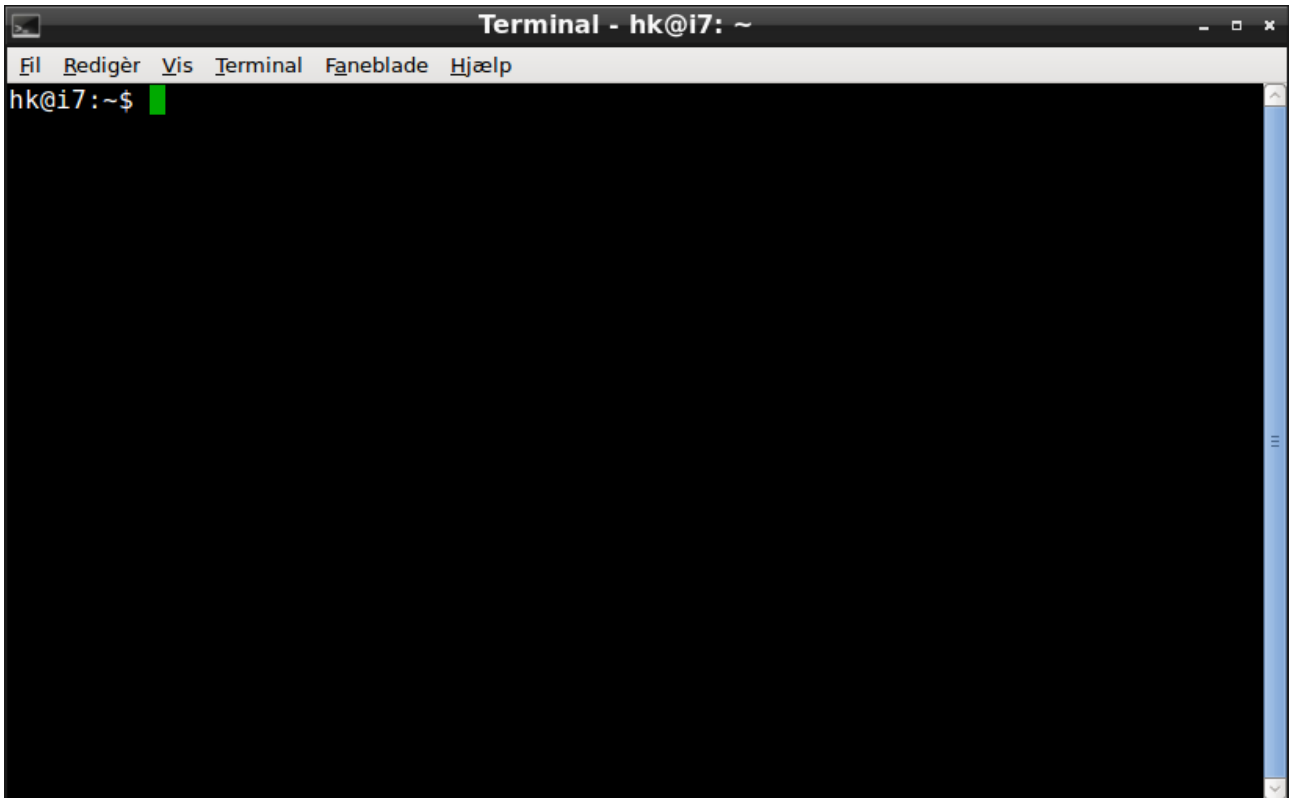


CLI for GUI'ister



Notat til brug for RPi øvelser

Henrik Kressner

August 2019

Indholdsfortegnelse

CLI For GUI'ister.....	2
Brug ls kommandoen.....	5
Brug nano editoren.....	6
Brug cp (copy) kommandoen.....	7
Brug cd (Change Directory) og pwd (Present Working Directory) kommandoen.....	8
Brug af Make Directory (mkdir) kommandoen.....	9
Afrunding.....	10
Eksempler.....	11

CLI For GUI'ister

Dette dokument er skrevet for GUI (Graphical User Interface) brugere, der vil baldre ruden og se ind bagved. Sagt på en anden måde, vi vil kigge ned i computerens maskinrum.

Brugerinterfacet på en computer kan i grove træk betragtes som på figuren til højre.

Inderst har vi HardWaren (HW), det er populært sagt, ”det man kan røre ved”.

For at starte computeren op når den tændes, er der et program der kaldes FirmWare.

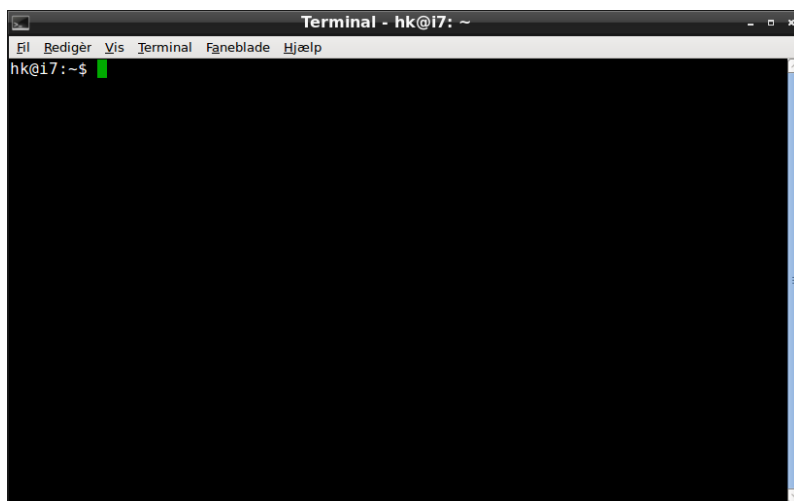
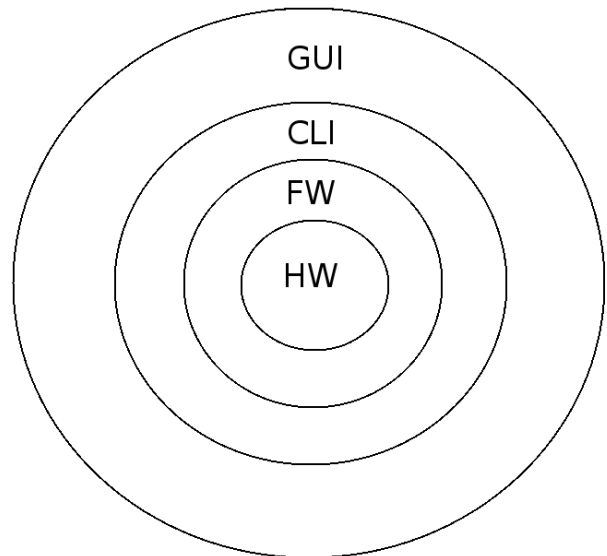
For at mennesker kan kommunikere med computeren er der skabt et lag uden om firmwared, kaldet Command Line Interface. (CLI)

I nogle systemer er der bygget et GUI uden om CLI'et, det er her de fleste mennesker befinder sig.

Det hele er altså bygget om som nogle skaller uden om hinanden, det kaldes også skaller, eller på engelsk shell. (Når vi kommer ned i CLI'et skal du tænke engelsk, selv om det er mere end 40 år gammelt, er det aldrig blevet oversat til et andet sprog.)

For at komme ned i CLI'et fra GUI, skal du bruge et program der kan have forskellige navne, det mest udbredte navn er terminal. (Et andet navn for et program er applikation, eller den mere moderne forkortelse, en app.)

Når du starter terminal programet ser det nogenlunde således ud:



Helt til venstre står promptet, det er forskelligt fra system til system, og afhænger af systemets konfigurationen. Det eneste du kan regne med er, det er der. Umiddelbart til højre for promptet står en markør og blinker, det kaldes cursoren.

Når du kommer ind i terminal programmet (shell'en) kan du godt glemme alt om at bruge musen, den er uden effekt. (Du kan dog stadig, samtidig, bruge musen som du plejer i GUI'et)

Når du kommer ned i CLI'et er de næsten ens, uanset om du er på en UNIX variant eller på Microsoft (MS), der er dog nogle få markante forskelle. (UNIX varianter er alt andet end MS systemer, hertil tæller alt Apple udstyr fabrikeret efter 1980, alt Android baseret udstyr, BSD, og alt der kalder sig noget med Linux.)

Hvis du vil vide mere om MS CLI, kan du fatte en søgemaskine og søge på:

windows tutorial cli

Hvis du vil vide mere om UNIX varianter CLI, kan du fatte en søgemaskine og søge på:

UNIX tutorial cli

Den største forskel er, MS ser ingen forskel i brugen af små og store bogstaver, alle andre gør. Hvis du gør det til en vane altid at betragte små og store bogstaver som forskellige, får du ingen problemer med det.

Ikke alle kommandoer er ens, men begge systemer har indbygget hjælp, i MS kan du taste `help` og trykke enter, i UNIX varianterne skriver du `man` (kort for manual) og trykker enter. (Du får også brug for at søge hjælp på nettet)

Nu drejer dette dokument sig om at kunne betjene en Raspberry Pi (RPi), der kører som en headles server, (Headles betyder i denne sammenhæng, ingen skærm/tastatur/mus)

For at logge (ansvarligt) ind på en headles server, skal vi bruge et program, og det er identisk, uanset om du bruger det ene eller det andet system, du skriver blot

```
ssh alfa@10.1.1.1
```

Under den forudsætning dit brugernavn er alfa, og IP adressen på den server du vil kontakte er 10.1.1.1

Herefter vil du blive anmodet om at indtaste dit password, (det bliver ikke skrevet på skærmen, der er ingen grund til at give en lurer den fordel), og trykke enter. Efter et øjeblik er du logget ind, (hvis du har stavet dit password rigtigt). Herefter taler vi kun UNIX varianter.

For at begå dig i de småøvelser vi laver, skal du kunne bruge følgende kommandoer:

ls	(LiSt indhold af directory)	cp	(Copy/Kopier)	
cat	(Vis indhold af en fil)	mkdir	(MakeDirectory)	
cd	(Change Directory)	pwd	(Vis hvor i directorysystemet du står)	og

hjelpeprogrammet nano, som er en editor. (En editor er en App der er skabt til at redigere i simple tekstfiler.)

Når du er logget ind på Rpi'en, er der et dollartegn der fungerer som et prompt, lige til højre for det står der en markør og blinker, her kan du taste kommandoer. (Der kan stå mere til venstre for dollartegnet, ignorer det for nuværende)

Brug ls kommandoen

ls kommandoen viser indholdet af det directory hvor du står:

```
$ ls [tryk enter]
$
```

Ovenstående betyder ikke directoryet (mappen) er tomt, der kan være skjulte filer, dem ser du ved at skrive:

```
$ ls -a
.  ..  .bash_logout  .bashrc  .profile
alfa@HS1:~ $
```

Et filnavn der starter med et punktum, er et skjult filnavn.

De to "filer" . (punktum) og .. (punktum punktum) er ikke "rigtige" filer, det er directory's. Hvis der står et (1) punktum er det det directory du står i, hvis der står to (2) punktummer er det directoryet lige over der hvor du står, du kan altså skrive:

```
$ ls ..
alfa  beta  ceta  deta  pi
```

Det du ser i dette tilfælde er directory's for alle brugere på det system du er logget ind på.

Du kan se en lang (long) udgave af directoryet ved at benytte argumentet l for long:

```
$ ls -l
drwxr-xr-x  2 alfa alfa 4096 dec 13 12:45 alfa
drwxr-xr-x  2 beta beta 4096 dec 13 12:45 beta
drwxr-xr-x  2 ceta ceta 4096 dec 13 12:45 ceta
drwxr-xr-x  2 deta deta 4096 dec 13 12:45 deta
drwxr-xr-x 19 pi   pi   4096 okt  9 09:57 pi
```

Vi vil ikke gå dybere ind i hvad ovenstående betyder (søg på nettet) men når der står et d helt til venstre på en linie, betyder det i denne sammenhæng, det efterfølgende er et directory.

Hint 1: Alle kommandoer afsluttes ved at trykke på enter tasten.

Hint 2: Husk mellemrum mellem kommando og argumenter.

Hint 3: Minustegnet (-) kaldes en switch,

Brug nano editoren

Du starter nano editoren ved at skrive:

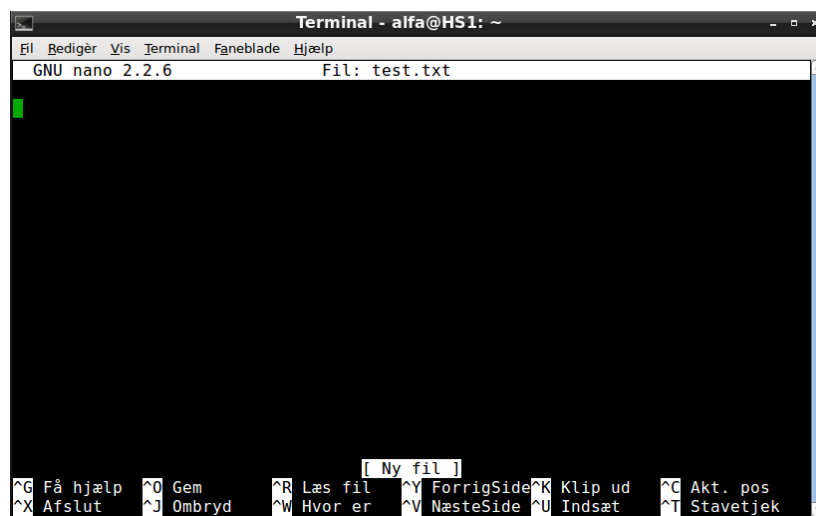
```
$ nano [filnavn]
```

Bemærk [filnavn] skal ikke tastet ind som det står, det beskriver et argument, og det argument skal være et filnavn, efterfulgt af et tryk på enter tasten.

Hvis du vil redigere i en fil ved navn test.txt skriver du:

```
$ nano test.txt
```

Når du har trykket på enter kommer følgende skærmbillede:



Øverst i midten står navnet på den fil du er ved at redigere, nederst er der 2 linier med hjælpetekst. Pil op ^ betyder tryk på Ctrl tasten, og hold den nede mens du taster den næste tast. (Dette kaldes en hotkey)

Hvis du eksempelvis skriver; Hallo verden og trykker ^X, skal du sige ja til at gemme og acceptere filnavnet, derefter står du ude i promptet. Du kan tjekke indholdet af filen ved igen at skrive:

```
$ nano test.txt
```

Når du står ude i promptet kan du liste indholdet af en fil ved at skrive:

```
$ cat [filnavn]
```

Uden at ”gå ind” i filen, altså uden at starte editoren.

Hint: En editor er en meget simpel udgave af et tekstbehandlingsprogram, det er velegnet til det vi kalder scripting.

Brug cp (copy) kommandoen

Du kan kopiere en fil ved at bruge cp på følgende måde:

```
$ cp [frafilnavn] [tilfilnavn]
```

Hvis du vil kopiere den fil du lavede i nano, kan du skrive:

```
$ cp test.txt ny.txt
```

Du kan teste om kopieringen er sket ved at skrive

```
$ ls  
ny.txt test.txt
```

Eller hvis du vil vide mere om filerne kan du sætte l (som i long) på:

```
$ ls -l  
-rw-r--r-- 1 alfa alfa 13 dec 16 09:50 ny.txt  
-rw-r--r-- 1 alfa alfa 13 dec 16 09:49 test.txt
```

Hvis du vil flytte din fil over til det directory hvor webserveren kører, og kalde den nye fil index.html, gøres det således:

```
$ cp test.txt /var/www/html/index.html
```

Du kan kontrollere kopieringen er sket ved at skrive:

```
$ ls /var/www/html
```

Eller med long udskrift:

```
$ ls -l /var/www/html
```

Hvis du kun vil se den fil du har kopieret kan du skrive:

```
$ ls /var/www/html/index.html
```

Eller på lang form:

```
$ ls -l /var/www/html/index.html
```

Hint 1: Ved kopiering sker der ingen ændring i den fil der kopieres fra. Den fil der kopieres til overskrives derimod, hvis den findes og du har rettigheder til det.

Hint 2: Filnavne kan kombineres med en path, søg på nettet hvis du vil vide mere om det.

Brug cd (Change Directory) og pwd (Present Working Directory) kommandoen

Du kan bevæge dig rundt i directory systemet med cd kommandoen. Hvis du vil se hvor du står kan du skrive:

```
$ pwd
/home/alfa
```

Her kan du se du står i alfa's login directory. (På alle UNIX varianter er det udbredt at hver bruger har sit eget virtuelle stisystem under /home directoryet.)

Hvis du vil bevæge dig op i directoryet lige over dig, kan du skrive:

```
$ cd ..
```

Fordi punktum punktum er en forkortelse for directoryet lige over hvor du står. Du kan i dette tilfælde i stedet skrive:

```
$ cd /home
```

Fordi home directory'et er placeret lige under roden, vist med /. I begge tilfælde vil svaret på at bruge pwd være det samme, nemlig:

```
$ /home
```

Hvis du på et tidspunkt bliver nægtet at skifte directory, vil det normalt skyldes du ikke har rettigheder til at kigge i det directory. Hvis du eksempelvis skriver:

```
$ cd /root
```

vil svaret blive noget i retning af:

```
bash: cd: /root: Adgang nægtet
```

På UNIX varianter der er sat korrekt op, (og det er din RPi) kan du altså roligt bevæge dig rundt i directorysystemet, du kan kun ødelægge noget for dig selv. Du har ikke rettigheder til se "hemmeligheder" og du har ikke rettigheder til at beskadige/slette filer du ikke må "pille" ved. Hvis du eksempelvis prøver at ændre en systemfil ved navn `resolv.conf` ved at skrive:

```
$ nano /etc/resolv.conf
```

Kommer du ind og kan se filen, men hvis du taster noget ind og forsøger at gemme, får du følgende fejlmeddelelse:

```
[ Fejl ved skrivning af /etc/resolv.conf: Adgang nægtet ]
```

Brug af Make Directory (mkdir) kommandoen

For at være sikker på du er "hjemme" (altså i dit hjem directory) starter du med at skrive:

```
$ cd
```

Når du bruger cd kommandoen uden argument, sender systemet dig automatisk til dit hjemmedirectory, uanset hvor du står.

Hvis di vil oprette et directory ved navn MitDir skriver du:

```
$ mkdir MitDir
```

Du kan se directoryet ved at skrive `ls` eller `ls -l`.

Du kan skifte ned i directoryet ved at skrive:

```
$ cd MitDir
```

Herefter kan du se dig om med `ls` kommandoen, du kan oprette redigere en fil med `nano` og du kan skifte directory `cd` kommandoen.

Afrunding

Du har nu viden nok, evt. med lidt hjælp fra en søgemaskine, til at bevæge dig frit rundt på en UNIX variant. Faktisk kan du (næsten) gøre det samme på et MS system, her er nogle af kommandoerne lidt anderledes, og der er stor fare for du ved et uheld, kommer til at beskadige systemfiler, og dermed "crashe" dit system, hvis du "piller" for meget.

På enhver UNIX variant kan du eksempelvis skrive:

```
$ cd /
```

Så lommer du op i roden af filsystemet, og kan kigge dig om med ls kommandoen

```
$ ls -l
```

Svaret udelades her, det er for langt, men alle linier der starter med et d, er et diretory, som du kan forsøge at skifte ned i og se dig om med ls. Du kan også forsøge at kigge i filen med nano, hvis du er nysgerrig, men forbered dig på overraskelser, (det kan se ud som om dit system "hænger", hvis det sker logger du bare ud og ind igen.)

Hint: På Windows systemer hedder roden ikke /, men \ (Bakslash) du skal altså taste cd \, for at komme op i roden på et MS system.

Eksempler

Eksempel 1:

Hvis du vil skifte over til webdirectoryet og kopiere filen test.txt derover, og give den det nye navn index.html, kan det se således ud:

```
$ cd /var/www/html
$ cp /home/alfa/test.txt index.html
```

Eksempel 2:

Hvis du er bruger alfa, og vil se hvad der ligger af filer i beta's hjemmedirectory, kan du skrive:

```
$ ls ../beta
```

Eller på lang form

```
$ ls -l ../beta
```

eller med fuld path:

```
$ ls /home/beta
```

Eksempel 3:

Hvis du vil kopiere filen test.html der ligger i deta's hjemmedirectory, til dit hjemme directory, kan du skrive:

```
$ cd
```

Det bringer dig til dit hjemmedirectory, uanset hvor du står. Derefter kan du kopiere filen til dit hjemmedirectory ved at skrive:

```
$ cp /home/deta/test.html .
```

Så bliver den kopieret over til dig, HVIS du har ret til at læse filen. Husk et enkelt punktum betyder "her hvor jeg står". Hvis du vil ændre filnavnet til min.html i kopieringsprocessen kan du skrive:

```
$ cp /home/deta/test.html min.html
```

Eksempel 4:

Hvis du vil oprette et directory ved navn tekst og kopiere filen test.html derved, skal du først oprette directoryet:

```
$ mkdir tekst
```

Derefter skal du kopiere filen derved ved at skrive:

```
$ cp test.html tekst/
```

Husk skråstregen til sidst, ellers får du en fejlmeddelelse. (Hvorfor mon?)