

## 3 - UNIX/Linux: Filosofi og Syntaks

---

# Vi - editoren



# Hvad er vi?

---

- **Vi** er en teksteditor der følger med alle Linux og UNIX distributioner.
- **Vi** betjenes normalt uden brug af mus
- **Vi** skal IKKE sammenlignes med et almindeligt tekstbehandlings program
- **Vi** bruges til at skrive/redigere rene tekstfiler (rent ASCII – format).
- De fleste ***funktioner aktiveres*** med ***et enkelt tegn***
- Her på kurset gennemgås de mest grundlæggende funktioner i **Vi**
- **SuperUsers** tilbyder ***et éndags-kursus*** i **Vi**, hvor der gennemgås langt flere faciliteter (SU-103).



# Hvad er vi?

---

Vi findes i mange kloner de mest kendte er  
*vim* og *gvim*

**Vi er *svær* at lære**  
***Let* at bruge!**



# Start af Vi editoren

---

**Vi editoren kan startes på følgende måder:**

```
$ vi          #(start af ny tom unavngivet fil)
$ vi nyfil    #(start af ny tom fil med navnet "nyfil")
$ vi glfil    #(åben og redigér gammel fil: "glfil")
```



# Vi-editorens tre tilstande

---

## Vi-editorens tre tilstande

1. Kommando tilstand
2. Indtastnings tilstand
3. Bundlinje tilstand

**Vi-editoren starter altid i ”*kommando tilstand*”**



# Vi-editorens tre tilstande

---

**De tre tilstande tilgås typisk ved i kommando tilstanden at taste:**

- 1. i (I kommando-tilstanden) = indtastnings tilstand**
- 2. : / ? (I kommando-tilstanden) = bundlinje tilstand**

**ESC = for at komme tilbage til kommando tilstand**



# Vi-editoren: Kommando-tilstanden

---

**Kommando-tilstanden bruges bl. a. til :**





- 1. At bevæge sig rundt i dokumentet ved hjælp af “flyttekommandoer”**
- 2. Klippe og kopiere i eksisterende tekster**
- 3. Slette tegn og linjer i eksisterende tekster**
- 4. Skifte til redigerings-tilstand og bundlinje tilstand**



# Vi-editoren: Kommando-tilstanden

---

## Kommando-tilstanden's flytte kommandoer

-  h Flytter cursoren 1 tegn til venstre
-  j Flytter cursoren 1 linje nedad
-  k Flytter cursoren 1 linje opad
-  l Flytter cursoren 1 tegn til højre





# Vi-editoren: Kommando-tilstanden

---

## Kommando-tilstanden's flytte kommandoer

Flyt cursor på den *aktuelle linje*:

- 0 Flytter cursoren til *første tegn* på linjen
- \$ Flytter cursoren til *sidste tegn* på linjen
- w Flytter cursoren til *næste ord* (mod højre)
- b Flytter cursoren til *forrige ord* (mod venstre)

Flyt cursor *sidevis*:

CTRL + f

Bladrer 1 side (skærm/vindue) *fremad* og flytter cursoren til *øverste linje*.

CTRL + b

Bladrer 1 side (skærm/vindue) *baglæns* og flytter cursoren til *nederste linje*.



# Vi-editoren: Kommando-tilstanden

---

## Kommando-tilstanden's SLETTE kommandoer

- dd** Sletter den aktuelle linje
- ndd** Sletter *n* linjer
- x** Sletter det tegn kurseren står på
- nx** Sletter *n* antal tegn mod højre

Når noget slettes med ovennævnte kommandoer, lægges det i en midlertidig buffer, der hører til dokumentet.

(Der gemmes altså en midlertidig kopi af det slettede)

- p** (lille p) indsætter det kopierede under den aktuelle linje
- P** (stort P) indsætter det kopierede over den aktuelle linje



# Vi-editoren: Kommando-tilstanden

---

## Kommando-tilstanden's kopierings kommandoer

- yy**      Kopierer den aktuelle linje
- nyy**     Kopierer *n* linjer
- x**        Stetter det tegn kurseren står på
- nx**      Sletter *n* antal tegn mod højre

Det kopierede lægges det i en midlertidig buffer der hører til dokumentet.

- p**        (lille p) indsætter det kopierede under den aktuelle linje
- P**        (stort P) indsætter det kopierede over den aktuelle linje

Indholdet af bufferen kan indsættes et ubegrænset antal gange, så længe der ikke klippes eller kopieres nye linjer ind i bufferen, og så længe vi ikke forlades.



# Vi-editoren: Kommando-tilstanden

---

## Kommando-tilstanden: Gentag, Fortryd og Join

- . Man kan gentage den sidste editering/ændring ( f.eks. indsætning eller sletning) ved at taste et punktum.
- u (undo) fortryder foudgående ændring
- J (Stort J) Join teksten i en linje med indholdet i den efterfølgende linje.



Lav øvelse 5.1, 5.2 og 5.3

# Vi-editoren: Indtastnings-tilstanden

---

## Ind og ud af indtastnings-tilstanden:

- i        Bruges som oftest når du er i kommando-tilstanden og ønsker at komme ind i indtastnings- tilstanden
- ESC    for at vende tilbage til kommando-tilstanden

## Indtastnings-tilstanden bruges til at skrive og redigere i tekster.

Hvis du *ikke er vant* til at benytte *kommando-tilstandens hurtige muligheder* for at navigere rundt i teksterne, kan det være *lettere at navigere* rundt og ændre i eksisterende tekster medens man er i *Indtastnings-tilstanden*.

Når man er i *indtastnings-tilstanden* kan man bruge computerens *almindelige klippebord* til at indsætte tekster der ligger uden for Vi. Det vil sige man faktisk har 2 buffere til rådighed, når man skal kopiere og indsætte tekster.

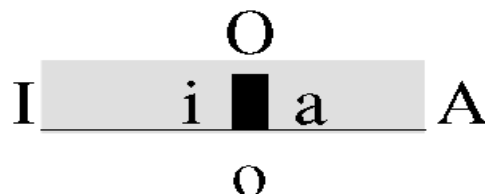


# Vi-editoren: Indtastnings-tilstanden

---

## Flere måder at komme i indtastnings-tilstanden:

- i** Gør klar til at tilføje tekst foran cursoren
- I** Gør klar til at tilføje tekst forrest på linjen
- a** Gør klar til at tilføje tekst efter cursoren
- A** Gør klar til at tilføje tekst bagest på linjen
- o** Indsætter en ny tom linje under aktuelle linje, så man kan tilføje tekst
- O** Indsætter en ny tom linje over aktuelle linje, så man kan tilføje tekst



# Vi-editoren: Indtastnings-tilstanden

---

## Indtastnings-tilstanden: Erstatning af tekster

**r** Gør klar til at erstatte tegnet inden i cursoren med ét nyt tegn.

Der vendes automatisk tilbage til Kommandotilstanden efter indtastning af det nye tegn.

**R** Gør klar til at erstatte/overskrive eksisterende tekst på en linje startende med tegnet under cursoren.



Lav øvelse 5.4

# Vi-editoren: Bundlinje-tilstanden

---

## Bundlinjetilstanden

Bundlinje-tilstanden bruges til at udføre “ex-kommandoer” også kaldet “kolon-kommandoer”

Bruges først og fremmest til at:

- Søg og erstat
- Gemme (w)
- Afslutte (q)

## Sådan kommer du ned i bundlinje-tilstanden:

:                      Bruges til at udføre en kommando (f.eks :w eller :q)  
/ og ?                Bruges til at søge efter en tekst-streng

## Sådan forlader du bundlinje-tilstanden:

**CTRL+c**

Eller ved at slette indholdet i linjen og trykke ESC





# Vi-editoren: Bundlinje-tilstanden

---

## Søgning efter en tekst

- /blue      Søger efter første forekomst af ordet “blue”
- ?blue      Søger BAGUD i teksten efter ordet “blue”
- n          Søger efter næste forekomst af det søgte ord
- N          Søger efter tidligere forekomster af ordet i teksten

## Søg og erstat

- :s/søgetekst/erstatningstekst/      Søger og erstatter første forekomst af ordet “søgetekst” i den aktuelle linje
- :s/søgetekst/nytekst/g      Søger og erstatter alle forekomster af det søgte ord i den aktuelle linje
- :%s/søgetekst/erstatningstekst/g      Søger og erstatter alle forekomster af det søgte ord
- :%s/[175]/8/g      Erstatte alle forekomster af 1, 7 og 5 med tallet 8



# Vi-editoren: Bundlinje-tilstanden

---

## Nogle vigtige "kolon-kommandoer":

- **:w *filnavn***            Gem et dokument første gang.
- **:w**                        Gem et allerede navngivet dokument.
- **:q!**                        Afslut UDEN at gemme.
- **:wq**                        GEM og AFSLUT
- **:x**                         GEM og AFSLUT

Ofte bruger man bare ZZ i kommando-tilstanden for at gemme og afslutte.

- **:set nu**                  Vis linjenumre
- **:set nonu**                Fjern visning af linjenumre



Lav øvelse 5.5, 5.6 og 5.7



## Vi-editoren: Bundlinje-tilstanden

---

### Indsæt indhold af en fil i dokumentet:

- `:r filnavn`** Indlæser og indsætter filen *filnavn* efter den aktuelle linje.
- **`:10r filnavn`** Indlæser og indsætter filen *filnavn* efter linje 10.
- **`:0r filnavn`** Indlæser og indsætter filen *filnavn* før linje 1 (efter linje 0).
- **`:$r filnavn`** Indlæser og indsætter filen *filnavn* efter sidste linje i dokumentet.



Lav øvelse 5.8 og 5.9

# Makroer i vi

---

## Makroer i vi-editoren med map og map!

**map** Bruges til makroer i kommando tilstand

**map!** Bruges til makroer i indtastningstilstand

Map kan bruges i et dokument ved at gå i kommandolinjen og skrive f.eks. :

```
:map <F2> :r lark <CR>  
#eller  
:map! £ Denne tekst vil jeg anvende mange gange
```

Hvis du vil slette en makro gøres det med:

```
:unmap  
:umap! £
```



# Gemme makroer til senere brug

## Man kan gemme makroer i sit hjemmekatalog

I filen “.vimrc” kan du gemme makroer til senere brug.  
Det kunne for eksempel være indhold du ønsker at bruge  
hver gang du opretter et shell-script  
(Vi kommer lidt mere ind på shell-script senere på kurset)

```
$ vi .vimrc #(Åbner opretter fil til vores makroer)
```

```
map <F2> i#!/bin/bash <ESC>  
map <F3> :r!echo "\# This script is made by: $USER" ;date "+ on \%A \%d. \%B \%Y " <ESC>kJo  
map! £ #####<CR>#<CR># Comment: <CR>#<CR>#####
```

