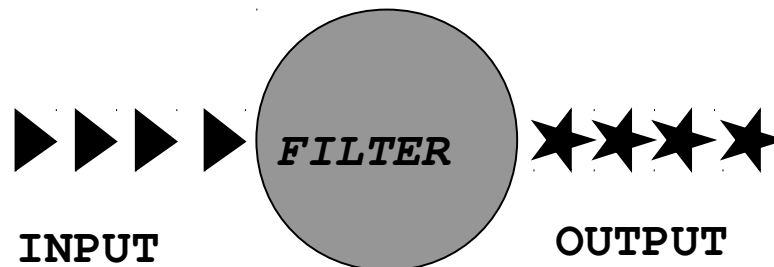


7 Håndtering af data og filer

Håndtering af Data og filer

7 Håndtering af data og filer

- **Data og filer** håndteres ofte af **kommandoer**, som ofte fungerer som **filtre**.
- Det er **små programmer**, der hver for sig er **eminent til en ting**
- Kort sagt er det **filtre** som:
 - Samler og læser data (**input**)
 - Gør et eller andet ved data
 - Afleverer data i tilpasset form (**output**)



7 Håndtering af data og filer

Om kommandoer og filtre

- Kommandoerne har ***korte navne*** skrevet med ***små bogstaver***.
- Kommandoer er ***brugervenlige***, ved ikke at gå i unødigt dialog med brugeren.
- **Et filter er et program der:**
 - Læser fra standard input (***stdin***)
 - Skriver til standard output (***stdout***)
 - Sender fejlmeddelelser til standard error (***stderr***)



7 Håndtering af data og filer

Oversigt over diverse kommandoer:

Visning

```
$ cat            #(Vis indholdet af en eller flere filer)
$ more/less     #(Vis indhold sidevis)
$ head          #(Vis 10 første linjer)
$ tail          #(Vis 10 sidste linjer)
```

Kopiering, flytning m.m

```
$ cp            #(Kopiér fil)
$ mv            #(Flyt/omdøb fil)
$ ln            #(Link til fil)
$ rm            #(Slet fil)
```

7 Håndtering af data og filer

Oversigt over diverse kommandoer *(fortsat)*:

Søgning

```
$ grep           # (Find linjer efter søgekriterie)
$ find          # (Find filer efter søgekriterie)
```

Sortering

```
$ sort           # (Sortér linjer)
```

Håndtering af dublet-linjer

```
$ uniq           # (Håndterer dubletter af linjer)
```

Håndtering af kolonner

```
$ cut           # (Håndterer kolonner i en fil)
```

7 Håndtering af data og filer

Oversigt over diverse kommandoer *(fortsat)*:

Sammenligning af filer

\$ cmp	<i> #(Sammenlign 2 binære filer)</i>
\$ diff	<i> #(Sammenlign 2 tekstfiler)</i>
\$ cksum	<i> #(Angiv checksum af fil)</i>
\$ md5sum	<i> #(Angiv md5 checksum af fil)</i>

Vis fil-karakteristika

\$ file	<i> #(Angiv type af fil)</i>
\$ strings	<i> #(Vis ASCIItekststreng i data -ex. wordfil)</i>
\$ wc	<i> #(Vis antal linjer, ord og tegn i data)</i>

Håndtering af data og filer



Brugeren kan selv vælge, hvor input skal komme fra:

- 1) Tastaturet (default)
- 2) Fil
- 3) Fra en anden proces (dvs. program).

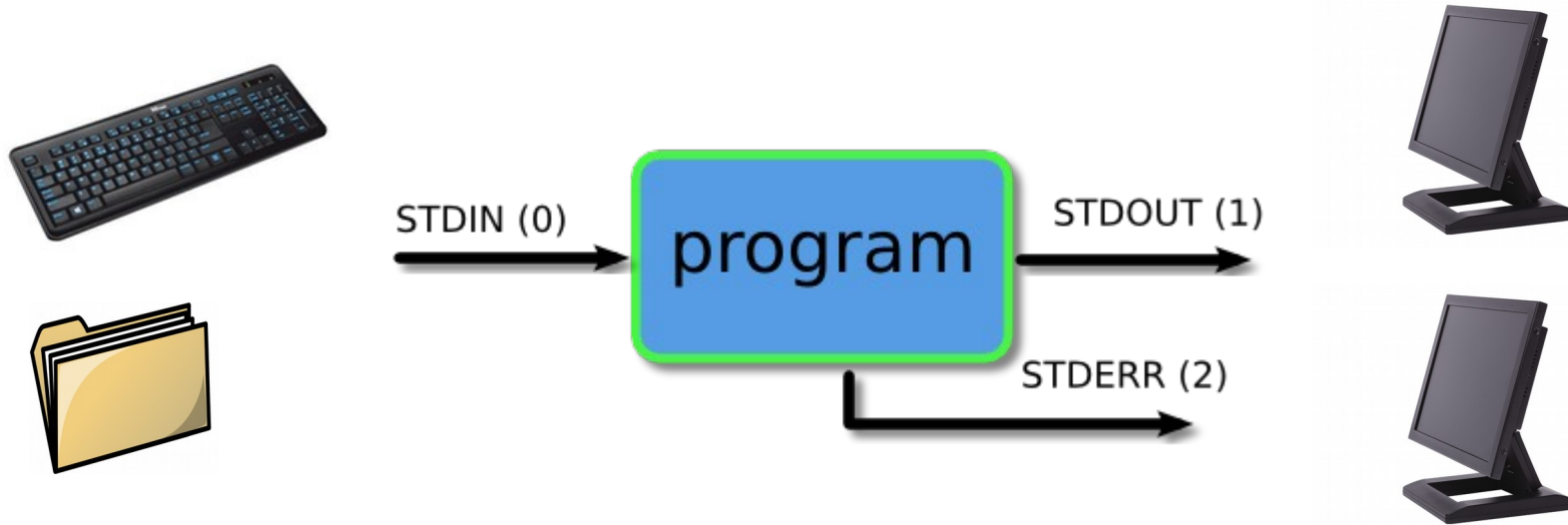
Brugeren kan selv vælge, hvor output skal hen:

- 1) Skærm (default)
- 2) Fil
- 3) Til en anden proces (dvs. program).

Brugeren kan selv vælge, hvor fejlmeddelelser skal hen:

- 1) Skærm (default)
- 2) Fil
- 3) Kan IKKE sendes til anden proces, men må omdirigeres

Håndtering af data og filer



STDIN (0)

```
< fil
cmd1 | cmd2
<< slut
```

STDOUT (1)

```
> fil
>>fil
>/dev/null
>&2
```

(output slettes)

```
&> /dev/null
```

STDERR (2)

```
2> fil
2>>fil
2>/dev/null
2>&1
```

(Mere om dette i Modul 11)

> er til omdirigering

& file-descriptor (uden det ville følgende 1 betragtes som et filnavn)

/dev/null er et "sort hul" hvor man kan vælge at sende output (1)/(2) til

7 Håndtering af data og filer

cat – Læg data i forlængelse af hinanden:

```
$ cat fil1 fil2
    #(fil1 og fil2 udskrives efter hinanden i stdout)
$ cat findesikke rim
    #(fejlmeldelse stderr og rim skrives efter hinanden i stdout)

$ cat -n fil1 fil2 #(udskriver linjenumre)
```

7 Håndtering af data og filer

Blad med more:

```
$ more /etc/passwd names
#(passwd og names udskrives efter hinanden en "side" ad gangen)
$ ls -l | more
#(Læg mærke til | "pipe" tegnet som sender output videre til "more")
```

Blad med less:

```
$ less /etc/passwd names
#(passwd og names udskrives efter hinanden en "side" ad gangen)
$ ls -l | less
#(Læg mærke til | "pipe" tegnet som sender output videre til "less")
```

7 Håndtering af data og filer

Head – vis begyndelsen af data:

```
$ head bibel           #(Her vises de 10 første linjer af "bibel")
$ head -23 bibel       #(Her vises de 23 første linjer af "bibel")
$ ls -l | head         #(Her vises de 10 første linjer af en liste)
```

Tail – vis slutningen af data:

```
$ tail bibel           #(Her vises de 10 sidste linjer af "bibel")
$ tail -23 bibel       #(Her vises de 23 sidste linjer af "bibel")
$ ls -l | tail +15     #(Her vises til og fra linje 15 af en liste)
$ tac fil1             #(Viser linjer i omvendt rækkefølge (UNIX tail -r))
```

◆ Lav øvelser Modul 7.1 (Løsningsforslag ligger efter øvelserne)

Håndtering af data og filer

Filnavne og links:

- Et **katalog** er en speciel **fil**, som indeholder en tekststreng for hver **fil** i kataloget samt et tal, **inoden**, som systemet bruger til at finde filens dataindhold.
- En **fil** kan godt være repræsenteret med flere forskellige *navne*, men **kun én inode**.
- Hvert **filnavn** med fælles **inode** kaldes for links. Og antallet vises med ls-kommandoen.
- Et **symbolsk link** er en **fil** som henviser til en **anden fils dataindhold**.

Man kan liste **inoderne** med:

```
$ ls -i  
eller  
$ ls -li
```

Håndtering af data og filer

cp - kopiering af filer:

- Kommandoen **cp** laver en *ny fil* med *sin egen inode*
- **Brugeren** som kopierer *skal have læseret til filen* og adgang til kataloget, som filen ligger i
- Den *nye fil ejes af brugeren*
- *Indholdet* er *det samme* som den *oprindelige fil*
- *Rettighederne og datoer* kan være anderledes end originalen
- *Eksisterende filer* med *kopinavnet* kan *blive overskrevet*.

```
$ cp num numkopi    #(Kopien får et andet navn end originale)
$ cp num /tmp/num    #(Kopien får et andet navn og placeres i /tmp)
$ cp num /tmp        #(Kopien får samme navn og placeres i /tmp)
$ cp num price size /tmp  #(3 filer kopieres og placeres i /tmp)
```

Håndtering af data og filer

cp – r kopiering af et katalog med indhold:

- Når Kommandoen **cp** tilføjes **flaget -r** kopieres indhold af et katalog ***recursivt***.
- Det vil sige ***alt indhold*** fra det oprindelige katalog kopieres med ***over i det nye katalog***

```
$ cp -r brevdir brevdir_kopi  #(kataloget oprettes samtidig)
```

Håndtering af data og filer

mv - flyt eller omdøbning af filer:

- Ændrer man **navnet på en fil**, er det **faktisk kataloget** man redigerer, da **kataloget** jo **indeholder tekststrengen** som er filens navn. **Inoden er uændret**
- **Flytter man** filen til **et andet katalog**, **redigerer man to kataloger**
- **Flytter man** filen **til** en **anden partition**, svarer det til en **kopiering og sletning**

```
$ mv num nylum          #(Filen num omdøbes til nylum)
$ mv num /tmp/nylum     #(Filen num flyttes /tmp og omdøbes til nylum)
$ mv num /tmp            #(Filen num flyttes til /tmp uden at skifte navn)
$ mv color price size /tmp  #(3 filer flyttes samtidig til /tmp)
```

Håndtering af data og filer

ln – link til fil(er):

- Oprettelse af et (ekstra) hard-link til en enkelt fil (dvs. opret en ekstra katalogindgang)
- De skal ligge i samme filsystem, da de deler *inode*
- Første argument er den eksisterende fil. Andet argument er det nye link eller destinationskataloget
- Kataloger kan IKKE hardlinkes

```
$ ls -li dogs          #(Læg mærke til filens(inode)/link:)  
# 2102656 -rw-r--r-- 1 bruger bruger 152 Oct 31 11:27 dogs  
  
$ ln dogs dogs_hot    #(dogs gives et extra link)  
  
$ ls -oli dogs dogs_hot #(Læg mærke til filens(inode)/link:)  
# 2102656 -rw-r--r-- 2 bruger 152 Oct 31 11:27 dogs  
# 2102656 -rw-r--r-- 2 bruger 152 Oct 31 11:27 dogs_hot
```


Håndtering af data og filer

In – link til fil(er) fortsat:

- Filen får et ekstra katalog

```
$ ln dogs dir/hund       #(dogs gives et extra link i kataloget dir)

$ ls -oli dogs dogs_hot  #(Læg mærke til filens(inode)/link:)
# 2102656 -rw-r--r-- 3 bruger 152 Oct 31 11:27 dir/hund
# 2102656 -rw-r--r-- 3 bruger 152 Oct 31 11:27 dogs
```

- Filen får et ekstra link med samme navn i kataloget *dir*

```
$ ln dogs dir           #(Fejler - dir er et katalog og opfattes derfor
                         ikke som et nyt filnavn)

$ ls -oli dogs dir/{dogs,hund}
# 2102656 -rw-r--r-- 4 bruger 152 Oct 31 11:27 dir/dogs
# 2102656 -rw-r--r-- 4 bruger 152 Oct 31 11:27 dir/hund
# 2102656 -rw-r--r-- 4 bruger 152 Oct 31 11:27 dogs
```

Håndtering af data og filer

In -s Symbolske link (*symlinks*):

- *Symbolske links* er *filer*, som indeholder en *reference* til en fil
- *Symbolske links* kan referere til *filer* i *andre partitioner*
- Også *Kataloger* kan *linkes symbolsk*
- Første argument er en streng som skal være navnet på referencefilen – med eller uden sti
- Man kan IKKE overskrive *symbolske links*.
(I givet fald må de slettes først og derfor oprettes igen)
- *Symbolske links* får deres egen *inode*

7 Håndtering af data og filer

Eksempler på symlinks:

```
$ ln -s cats symcats
    # (Opretter et symbolsk link, symcats, til filen cats)

$ ls -li cats symcats
    # (Læg mærke til de 2 filer får forskellig inode m.m.):
2102629 -rw-r--r-- 1 bruger bruger 121 Oct 31 14:26 cats
2097935 lrwxrwxrwx 1 bruger bruger   4 Nov 14 14:07 symcats -> cats

$ cd dir ; ln -s ../cats symcats2
    # (Skifter til kataloget dir og opretter et symbolsk link symcats2
    til filen cats i forældre-kataloget)
```

Håndtering af data og filer

rm – sletning af links:

- *rm* sletter angivne links
- Filer kan slettes hvis man har *skrive-ret* til kataloget
- Har man ikke skrivetilladelse til filen, advares man
- En fils data er utilgængelig, når dens sidste hardlink er slettet

```
$ rm brev           #(Sletter link i arbejdskataloget)
$ rm /etc/passwd    #(Sletter link i /etc - hvis man må!)
$ rm -i hund        #(-i = Der skal kvitteres for sletning)
$ rm -f size         #(Der kommer ingen advarsel)
$ rm -r dir          #(PAS PÅ - dir slettes med ALT indhold)
```



Lav øvelse 7.2

Håndtering af data og filer

grep – søg i datalinjer:

- *grep* gennemsøger data for linjer, der indeholder et angivet *søgeudtryk*.
- Søgeudtrykket kan være almindelig tekst eller *regulære udtryk* (specielle tegn).
- De fundne linjer udskrives til *stdout*.

Eksempler:

```
$ grep ALFA fil1          #(Søger efter tekststrengen ALFA i fil1)
                           #(som standard er søgning case-sensitiv)
$ grep herren bibel
$ grep HERREN bibel
```

Håndtering af data og filer

Flag til grep:

Der findes en række forskellige flag til grep, hvoraf de vigtigste er

- c Resultatet er antallet af linjer, der opfylder søgeudtrykket
- v Resultatet er alle de linjer, der ikke opfylder søgeudtrykket
- n De fundne linjer udskrives med deres linjenummer forrest på linjen
- l Resultatet er navnene på de filer, der opfylder søgekriteriet
- i Ignore case –behandler små/store bogstaver ens
- r Foretag søgning rekursivt gennem underkataloger

```
$ grep -c HERREN * #(Antal forekomster af HERREN i pwd)
$ grep -n ALFA alfa #(viser linjenumre for de fundne linjer)
$ grep -i alfa alfa #(Ignorerer store/små bogstaver)
$ cp
```



Lav øvelse 7.3

find kommandoen – søger i kataloger og inode-tabel

- Kommandoen ***find*** søger efter ***filer*** i filtræet
- Der søges ud fra angivne søgekriterier (default: alle filer)
- Søgningen er afgrænset i henhold til ***katalog-rettighederne***
- Søgekriterierne kan angives med ' ***quotes*** ' / " ***doublequotes*** "
- Brug ***quotes*** for ikke at tolke ***wildcards tegn*** i teksten
- Der kan søges på alle tilgængelige oplysninger i ***inode-tabellen*** eller ***kataloger***

Håndtering af data og filer

find kommandoen – syntax:

`find [søg herfra] [søgekriterium] [handling]`

`[søg herfra]` Katalog(er) udgangspunktet for søgning (default i Linux).

`[søgekriterium]` Filnavne eller inodeoplysninger.

`[handling]` Hvad der skal ske med de fundne filer.
Default vises filnavnet.

```
$ find ../ -name brev  #Søger efter filer med navnet brev
                        i forældrekataloget og underkataloger
$ find /home -name 'brev*'  #(Søg efter filer i /home,
                           som har "brev" i starten af navnet)
```


Håndtering af data og filer

find kommandoen – eksempler:

```
$ find .. -name 'brev*'      #(Søger efter filer i forældrekataloget,  
                             som har "brev" i starten af navnet)  
$ find / -user bruger       #(Søger efter ALLE filer ejes af bruger)  
$ find brevdire -mtime -1   #(Søger efter filer i brevdire der er  
                             modificeret inden for det sidste døgn)  
$ find ./ -name 'fil[1-3]' -exec cat {} \;  
                             #(Søger efter fil1, fil2 og fil3, og viser indholdet  
                             Bemærk det efterfølgende -exec flag og cat{} \;)  
$ find brevdire -name brev -ok rm {} \;  
                             #(Søger og sletter alle filer der hedder brev  
                             Bemærk det efterfølgende -ok flag og rm{} \;)  
$ find ./ ! -user bruger2
```

Håndtering af data og filer

locate kommandoen

– en hurtig måde at finde en fils placering:

```
$ locate 'bibel'      #(Søger efter filer i hvor ordet bibel indgår  
                      i filnavnet ~ '*bibel*')  
$ locate -b '\bibel'  #(Søger efter filer hvor navnet eksakt matcher  
                      ordet bibel)
```

OBS!

Kommandoen er så hurtig, fordi den læser i databasen "mlocate.db" som ikke er opdateret hvis man ikke forud har kørt kommandoen:

```
$ sudo updatedb      #(kan kun afvikles med sudo/root rettigheder)
```



Lav øvelser 7.4

Håndtering af data og filer

**sort kommandoen –
sorterer efter ASCII-værdier og på linjebasis**

SYNTAX: sort [flag] [filnavn ...]

Der findes en række forskellige flag til sort kommandoen:

- b** Ignorer dubletter af skilletegn mellem felter
- r** Sortér i faldende rækkefølge
- f** Sortér store og små bogstaver som værende ens
- u** Udskriv kun identiske linjer 1 gang
- o** outfil Resultatet (output'et) skrives i outfil,
der godt kan være identisk med inputfil.
- k** tal[type] Sorter efter "tal'te" felter (ex. *kolonne nr.*) 3n
efter datatype [type] som f.eks. kan være:
 - r** - reverse
 - n** - numerisk

Håndtering af data og filer

Sort kommandoen

```
$ cat gurues           #(cat=usorteret)
  Bill   Joy          3      oct
  James  Bond         007    mar
  Ken    Thompson    2      sep
  Linux  Thorvald    03     dec

$ sort -k2 gurues  #(sorterer efter kolonne 2)
  James  Bond        007    mar
  Bill   Joy         3      oct
  Ken    Thompson    2      sep
  Linux  Thorvald    03     dec

$ sort -k3n gurues  #(sorterer efter kolonne 3 numerisk)
  Ken    Thompson 2      sep
  Bill   Joy      3      oct
  Linux  Thorvald 03     dec
  James  Bond     007    mar

$ sort -k4rM gurues  #(Efter kolonne 4 omvendt-månedsværdi)
  Linux  Thorvald   03      dec
  Bill   Joy        3      oct
  Ken    Thompson   2      sep
  James  Bond       007     mar
```

Håndtering af data og filer

uniq kommandoen – håndtering af dubletlinjer i sorterede data

SYNTAX: *uniq* [flag] [inputfil] [outputfil]

- *uniq* Udskriver inddata, men udelader dubletlinjer
- Dubletlinjer er linjer, som er identisk med den foregående linje
- Bemærk: Hvis der til *uniq* angives to filer som argument, vil den sidstangivne overskrives med *output*

flag:

- u** Udskriv kun unikke linjer.
- d** Udskriv linjer der er dubleret (men ikke dubletterne).
- c** Alle linjer udskrives én gang.
Forrest på linjen skrives, hvor mange forekomster, der er af linjen.
- s n** Undlad at sammenligne de første *n* tegn
- f m** Undlad at sammenligne de første *m* felter

Håndtering af data og filer

uniq / sort kommandoen

```
$ cat dublin   #(viser oprindelig fil med cat)
```

```
    alfa  
    beta  
    alfa  
    gamma
```

```
$ sort -o dublin dublin   #(filen sorteres og overskrives)
```

```
$ cat dublin
```

```
    alfa  
    alfa  
    beta  
    gamma
```

```
$ uniq dublin   #(viser kun dublerede linjer een gang )
```

```
    alfa  
    beta  
    gamma
```

Håndtering af data og filer

uniq kommandoen fortsat

```
$ cat uniqlin
  alfa
  alfa
  beta
  gamma
$ uniq -c uniqlin
  2 alfa
  1 beta
  1 gamma
$ uniq -d uniqlin
  alfa
```

Håndtering af data og filer

Håndtering af kolonner med "cut"

SYNTAX: `cut [flag] [inputfil]`

flag:

- f** [x,y] Udskriv kun den eller de kolonne numre.
- d** separator - I stedet for "TAB": Angiv en anden separator
f.eks. komma eller semikolon

```
$ cut -f3 gurues
# udskriver 3die kolonne i filen gurues

$ cut -d', ' -f6,8,9,12,13 Downloads/adresser
# Udskriver kolonne nr. 6,8,9,12 og 13 fra en
kommasepareret fil*
```

*) <https://download.aws.dk/adresser>



Lav øvelser 7.5 og 7.6

Håndtering af data og filer

cmp – Sammenligning af filer:

- **cmp** sammenligner to filer og udskriver *linjenummer* og *tegnnummer*, hvor filerne første gang afviger fra hinanden
- **cmp** bruges ofte til sammenligning af *binære filer*
- **cmp** kan også bruges til sammenligning af *tekstfiler* – specielt i de situationer, hvor man ikke er interesseret i detaljer men ***blot ønsker at få at vide, om filerne er ens, ja eller nej***

```
$ cmp dublin uniqlin      #(sammenlign dublin  og uniqlin)
                          # dublin uniqlin differ: byte 6, line 2
```

Håndtering af data og filer

diff og sdiff – Sammenligning af filer:

- **diff** viser et sæt koder (til ed / ex editoerer) til hvordan man kan gøre de to filer ens
- **sdiff** stiller de 2 filer op ved siden af hinanden og viser forskelle ved hjælp af tegnene:
 - | Linjer med forskelle
 - > Linjer der kun forekommer i fil2
 - < Linjer der kun forekommer i fil1

```
$ diff dublin uniqlin          #(viser koder med forskelle)
$ sdiff dublin uniqlin         #(filer vises ved siden af hinanden)
$ sdiff -s dublin uniqlin      #(her vises kun linjer med forskelle)
```

Håndtering af data og filer

Cksum og md5sum –

Tjek at filer er ens og ikke er ændrede:

- Typisk vil man gerne tjekke om indholdet af en fil /program svarer til det man forventer og filen f.eks ikke er blevet ændret eller beskadiget i forbindelse med download.
- Ofte ser man udgivere af et program oplyser **md5sum** i forbindelse med, at man kan downloade programmet.

```
$ cksum a.pdf b.pdf
$ md5sum a.pdf b.pdf
$ sha1sum a.pdf b.pdf   #(Her har vi manipuleret med filerne)
 #(Sikkerhedsstandarden for ovennævnte er for svag og kan manipuleres)
$ sha224sum a.pdf b.pdf
$ sha256sum a.pdf b.pdf
```

Håndtering af data og filer

file – filtypebestemmelse:

```
$ file bat brev dir                #(Viser filtyper for de 3 filer)
    bat:  ASCII text
    brev: ISO-8859 text
    dir:  directory
$ file ./* | grep text             #(Viser alle filtyper af typen text)
```

strings – vis ASCII-tekststrengene (binære som tekst):

```
$ cat /usr/bin/find                #(Viser binær fil)
$ strings /usr/bin/find            #(Viser kun ASCII tegn
                                   fra den binære fil)
```

Håndtering af data og filer

wc – tæl linjer, ord og tegn i data

- **WC** udskriver antallet af linjer, ord og tegn i en eller flere filer. *wc* er en forkortelse af *word count*.

```
$ wc kina rim  #(viser antal linjer, ord og tegn i de to filer)
  5  26 145 kina
  4   7  38 rim
  9  33 183 total

$ wc -l cats  #(viser antal linjer i filen cats)
$ wc -w cats  #(viser antal ord i filen cats)
$ wc -c cats  #(viser antal tegn i filen cats)
```



Lav øvelse 7.7, 7.8 og 7.9