

# Brukarrettleiing for UA

Gisle Sælensminde  
Hans Georg Schaathun  
Institutt for Informatikk  
Universitas Bergensis

9. september 1998

©1998 Hans Georg Schaathun <georg@ii.uib.no> og Gisle Sælensminde  
<gisle@ii.uib.no>; Institutt for Informatikk, Universitetet i Bergen.

Det er strengt forbudt å kopiera heile eller deler av denne handboka utan  
skriftleg løyve frå forfattarane eller instituttet.

Spørsmål tilknytta handboka kan rettast per epost til <manual@ii.uib.no>.

Boka er skriven i L<sup>A</sup>T<sub>E</sub>X2 $\epsilon$  og me har forbereda han for omsetjing med  
L<sup>A</sup>T<sub>E</sub>X2HTML.

# Innhold

<b>1</b>	<b>Det fyrste biletet</b>	<b>7</b>
1.1	Terminalar og innlogging . . . . .	7
1.1.1	Brukarnamn og passord . . . . .	7
1.1.2	Korleis logge inn . . . . .	7
1.2	Bruk av CDE . . . . .	7
1.2.1	Menyfunksjonar . . . . .	8
1.2.2	virtuelle skjermar . . . . .	8
1.3	Terminalvaktordninga . . . . .	8
1.3.1	Terminalkultur, køordning . . . . .	9
<b>2</b>	<b>Kommandolina og filsystemet</b>	<b>11</b>
2.1	Kommandolina . . . . .	11
2.1.1	Nokre enkle kommandoar (echo, date) . . . . .	12
2.2	Filsystemet . . . . .	12
2.2.1	Treet . . . . .	12
2.2.2	Kvar er me (pwd) . . . . .	13
2.2.3	Innhald i katalogen (ls) . . . . .	13
2.2.4	Flytta seg i filsystemet (cd) . . . . .	14
2.3	Filkommandoar . . . . .	15
2.3.1	Nye katalogar (mkdir, rmdir) . . . . .	15
2.3.2	Flytta filer (mv, rm, cp) . . . . .	15
2.3.3	Visa filer (less, cat) . . . . .	16
2.3.4	Utskrift (lpr, lprm, lpq, enscript) . . . . .	17
2.3.5	Avbryta eit program (Ctrl-C) . . . . .	18
2.4	Tilgangar . . . . .	18
2.4.1	Endra tilgangar (chmod) . . . . .	19
<b>3</b>	<b>Rettleiinger</b>	<b>21</b>
3.1	Unix-manual på maskina . . . . .	21
3.1.1	man, apropos, whatis . . . . .	21
<b>4</b>	<b>Dei fyrste programma</b>	<b>23</b>
4.1	Redigeringsprogram . . . . .	23
4.1.1	vi, vim . . . . .	23

<b>5</b>	<b>Internet</b>	<b>27</b>
5.1	Internet Generelt	27
5.1.1	Kva tenester finns på Internet	27
5.1.2	Epost generelt	27
5.1.3	Epostadressar	27
5.1.4	Hovud og kropp	28
5.1.5	dtmail, funksjonatitet, sende og ta imot epost	28
5.1.6	Lese epost	28
5.1.7	Filer som vedlegg(attachements)	28
5.1.8	mailx/UCB mail	29
5.1.9	Andre postprogram på instituttet	30
5.1.10	mailx/UCB mail	31
5.1.11	Postlister: kurslister, sstud, hstud, majordomo	32
5.2	WWW	33
5.2.1	Netscape	33
5.2.2	Kva er ein URL	34
5.2.3	Sidene til instituttet, kursinformasjon	34
5.2.4	Korleis lage egne heimesider	34
5.3	Usenet News	34
5.3.1	Kva grupper finns	34
5.3.2	Ein newslesar: slrn	34
5.3.3	Netiquette på Usenet	35
5.3.4	Ftp	35
5.3.5	Hente filer frå ein annan maskin	35
5.3.6	Anonym FTP, hente offentleg tilgjengelege data	36
5.3.7	Andre FTP-program	36
5.4	Søkemotorar	36
5.4.1	Sidene til Institutet	36
5.4.2	Boksøk : bibsys	37
5.4.3	Generellt www-søk: Altavista	37
5.4.4	Emnetalogar: <a href="http://www.Yahoo.com">http://www.Yahoo.com</a>	37
5.4.5	Ftp-søk: <a href="http://ftpsearch.ntnu.no">ftpsearch.ntnu.no</a>	37
5.4.6	Søk etter usenetartiklar: Dejanews	37
<b>6</b>	<b>Meir arbeid på kommandolina</b>	<b>39</b>
6.1	Ymse program	39
6.1.1	Sova (sleep)	39
6.1.2	Omtale	39
6.1.3	Kalender (cal)	39
6.2	Utdata og inndata (< >   (! less))	40
6.2.1	Utdata til fil (>)	40
6.2.2	Utdata til eit program	41
6.2.3	Inndata frå fil	41
6.3	Oppgåvekontroll	42
6.3.1	Avbryta eller stoppa ein kommando (Ctrl-Z, Ctrl-C)	42
6.3.2	Bakgrunnsprosessar (&)	42
6.3.3	Oppgåver (jobs, bg, fg)	42
6.3.4	Prosessar (ps)	43
6.3.5	Drepa prosessar (kill)	45
6.4	Filhandtering	46

6.4.1	Komprimering (gzip, compress, tar, gtar) . . . . .	46
6.4.2	Program i fleire versjonar (where) . . . . .	47
<b>7</b>	<b>Programering og kompilatorar</b>	<b>49</b>
7.1	Kva er eit programmeringspråk og kva er ein kompilator . . . . .	49
7.2	Make . . . . .	49
<b>8</b>	<b>Eige oppsett</b>	<b>51</b>
8.1	Korleis overstyre CDE med eige oppsett . . . . .	51
8.1.1	Kva oppstartsfiler finns. Likskap med satsfiler . . . . .	51
8.1.2	Kva filer skal ein ta utgangspunkt i . . . . .	51
<b>9</b>	<b>Reglement for Terminalbruk</b>	<b>53</b>
9.1	Tanken bak reglementet . . . . .	53
9.1.1	Netiquette - sunn bruk av internett . . . . .	53
9.1.2	Terminalromreglar . . . . .	53
9.2	Instituttet sitt reglement . . . . .	53
9.3	Universitetet sitt reglement . . . . .	53



# Kapittel 1

## Det fyrste biletet

### 1.1 Terminalar og innlogging

Undervisningsanlegget(UA) finn du i underetasjen på auditoriefløyen, Allegaten 55. Her finn du 40 Sun ultra 1 arbeidsstasjoner. Det er desse maskinene dei fleste kursa nyttar seg av i undervisninga. I tillegg nyttar ein del kurs fakultetet sine datamaskiner på faklaben. Faklaben finn du i første etasje på realfagbygget(etasjen over inngangen), i gangen som går mot studentsenteret.

#### 1.1.1 Brukarnamn og passord

Før du kan logge inn trng du eit brukarnamn på undervisningsanlegget, og passord. Dette får ein til vanleg på første gruppe i grunnkurs i i110, i160 og i120 og i001. For andre kurs må ein søke om brukar. Dette gjer ein på eigne skjema på instituttkontoret.

#### 1.1.2 Korleis logge inn

Når ein har fått tak i brukarnamn og passord, er neste steg å logge inn. Finn ein ledig maskin. Da har du eit vindauge som ber om 'login name' Her skriv du inn brukarnamnet ditt, trykk return-tasten. Etter det skriv du inn passordet ditt. Dersom både deler er rett, vil maskinen logge deg inn.

### 1.2 Bruk av CDE

Vindusystemet som køyrer på unixmaskinene er X-windows. X-windows treng ein vindustyreprogram, og her køyrer vi CDE. Det finns og andre vindustyrarar. Det er CDE som avgjer korleis skjermen skal sjå ut, og korleis mus og tastaturfunksjonane skal vere.

Når du startar opp Vil det sjå slik ut:

Nede er det ei verkøylinje, som ein kan starte program frå. Blant anna er det eit brevsymbol for å starte epostlesar, og eit pennesymbol for å starte teksteditoren. Dersom du trykker på desse med venstre mustast, vil du starte desse progamma.

### 1.2.1 Menyfunksjonar

Det er to typar menyar som du kan starte programm frå i CDE, rullemenyar og musmeny. Rullemenyen kjem fram dersom du trykk på eit av pilsymbola på vertøylinja.

Dersom du trykk på skjermssymbolet som kjem fram etter du har fått fram rullemenyen, startar det opp ein terminal. Fleire andre program kan startast opp på denne måten.

#### **musmeny**

Det går og å få fra menyar med å trykke på ein av mustastane over ein stad utan vindaug. Da vil det kome fram ein meny, som ser ut slik som under. Det er ein slik meny for kvar mustast

#### **Vindaugemeny**

Oppe i venstre hjørne på alle vindaug og program, er det et symbol. Dersom du trykker venstre mustast over det symbolet, kjem det fram ein meny. Denne menyen nyttast til funksjonar som har direkte med vindaug et å gjere. Blant anna har du desse funksjonane:

**Minimize** - Gjer vindaug et om til eit lite symbol (ikon), slik at det ikkje tar så stor plass på skjermen. Dersom du trykker på symbolet med venstre mustast, vil det bli eit vanleg vindaug et att.

**Close** - Avslutter programmet som køyrer i vindaug et.

**Raise** - Gjer at dette vindaug et kjem framst.

**Hide** - Legg dette vindaug et bak alle dei andre

### 1.2.2 virtuelle skjermar

CDE har såkalla virtuelle skjermar, som vil sei at ein kan få eit nytt skjembilete ved å trykke på eit av felta 1, 2, 3 eller 4 på vertøylinja. Det går an å ha ulike vindaug et på kvar av desse skjermane. På den måten kan ein få plass til fleire, utan at dei dekker over kvarandre.

## 1.3 Terminalvaktordninga

På UA er det terminalvaktordning. Terminalvaktene kan gi svar på spørsmål som har med bruken av maskinane å gjere, og faglege spørsmål i grunnkurset I110, I160 og I001.

Det er terminalvakt minst 2 timar dagleg. Oversikt over når du kan finne terminalvakter finn du på døra inn til store terminalrom. Terminalvaktene er enten på eit av terminalromma, eller i rommet som er innanfor laserskriveren.

Det går og an å sende epost til terminalvaktene med spørsmål. epostadressa til terminalvaktene er «terminal@ii.uib.no»

### **1.3.1 Terminalkultur, kørdning**

Det er ofte fullt på terminalromma. Derfor har dei som skal lage obligatorisk oppgåve rett til terminalane før dei som ikkje skal gjere noko fagleg. Det kan og hende at eit av terminalromma er reservert for gruppeundervisning, og dei som ikkje er med i desse gruppene, må forlate terminalrommet. Det er oppslag ved inngangen til store terminalrom om når terminalromma er reservert.

Meir detaljerte reglar for bruk av Instituttet sine datamaskiner finn du i eit kapittel seinare i heftet.



## Kapittel 2

# Kommandolina og filsystemet

### 2.1 Kommandolina

Det viktigaste verktøyet me har under Unix er kommandolina eller skalet («the shell»). På UA bruker me eit skal som heiter `tcsh`, «the Tenex shell».

Når du startar eit kommandovindauga (dtterm, xterm) ser du eit *klarteikn* («prompt»):

```
t04@bille 1 %
```

Klarteiknet viser kven du er og kva maskin du bruker. Talet til slutt er kommandonummeret. Maskina bille ventar no på den fyrste kommandoen fra meg, t04.

Når markøren blinkar bak klarteiknet kan du skriva kommandoen din, ordren din til maskina. Gjør han det ikkje, arbeider maskina med noko og du må venta.

Du kan bruka slettetastane og piltastane for å gå tilbake og retta, og når du har fått alt riktig, trykkjer du vognretur (return, enter). Prøv til dømes:

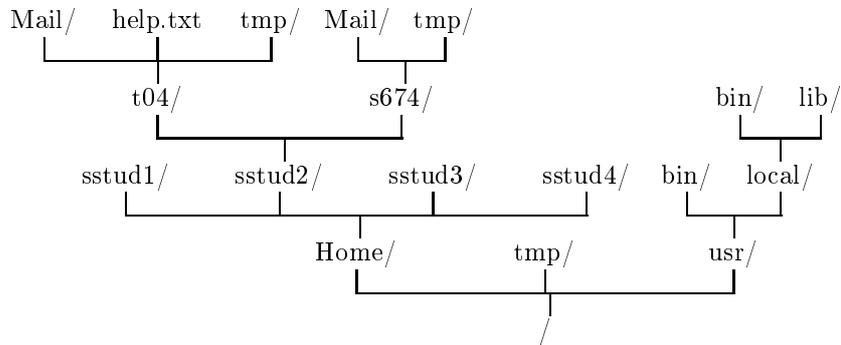
```
t04@bille 1 % echo Hello, world
Hello, world
t04@bille 2 % date
Tue Jul 28 18:38:22 MET DST 1998
t04@bille 3 % sate -u
sate: Command not found.
t04@bille 4 % date -u
Tue Jul 28 16:51:17 GMT 1998
t04@bille 5 %
```

Når du har eit par kommandoar bak deg kan du bruka piltastane opp og ned for å henta dei og bruka dei om att.

I eksempelet over er `echo` ein kommando eller eit program (skilnaden er litt diffus og ikkje så viktig), medan «Hello, world» er argumenta til `echo`. Den andre kommandoen me køyrer tar ingen argument. I den tredje lina har eg skrive feil, maskina kjenner ikkje til nokon kommando «sate». Til slutt har eg gjort det riktig; eg bruker den same kommandoen som i line nummer 2, men eg gjev ein opsjon i tillegg. Opsjonar byrjar alltid med ein minus (-) og «-u» her gjev meg tida som GMT i staden for den lokale tida. Kommandoar som både tek opsjonar og argument skal som regel ha opsjonane fyrst.

Når du bruker fleire opsjonar kan du som regel velja om du skriv dei saman, «-ab», eller kvar for seg, «-a -b». «ls -lF» og «ls -l -F» (sjå avsnitt 2.2.3) gjør nett det same.

Korleis starta dtterm ?



Figur 2.1: Ein del av filtreet på UA.

### 2.1.1 Nokre enkle kommandoar (echo, date)

#### Syntaks

```
date [-u]
echo [argument ..]
```

#### Omtale

Me har allereie sett døme på desse to programma.

`echo` skriv alle argumenta ut på skjermen. Han skriv eitt mellomrom mellom kvart argument.

`date` fortel kva dag og tid det er. `date -u` gjev tida GMT.

#### Døme

```
t04@skarabe 1 % echo Hello, world
Hello, world
t04@skarabe 2 % echo Hello,   world
Hello, world
t04@skarabe 3 %
```

## 2.2 Filsystemet

Når me arbeider med maskina hugsar han kva me gjør og tek vare på arbeida våre, men om me skal taka vare på dokument eller data til neste gong, må me lagra dei i *filsystemet*.

### 2.2.1 Treet

Filsystemet er organisert som eit tre (sjå figur 2.1). Nedst er rota som me kallar «skråstrek» («slash», «slask») og skriv «/». Allereie rett over rota tar fleire greiner av. Greinene kallar me katalogar. Greina «Home» er den viktigaste akkurat no, for der ligg heimekatalogane, og alle brukarane har kvar sin heimekatalog. Fra «Home» er der endå fleire greiner, eller underkatalogar, til dømes «sstud1», «sstud2», «sstud3» og «sstud4». Dei som har *s*-konto (eller *t*-konto), dvs. eit

brukarnamn med ein *s* (eller *t*) og eit tal, har heimen sin i ein av desse katalogane.

## 2.2.2 Kvar er me (pwd)

### Syntaks

```
pwd
```

### Omtale

pwd fortel oss kvar i treet me er.

```
t04@bille 5 % pwd
/Home/sstud2/t04
t04@bille 6 %
```

Det fulle namnet (eller stigen) til katalogen byrjar med ein skrâstrek for å markera at me går heilt ned til rota. Så skriv me kvar grein me skal klatra og skil kvar grein me skrâstrek.

«/Home/sstud2/t04» er heimekatalogen min, til t04.

Me kan òg bruka kortnamn (relativ stig). Kortnamna byrjar ikkje med nokon skrâstrek, og dei fortel kvar du skal gå fra der du er. Tenk til dømes at der er ein katalog «/Home/sstud2/t04/i110». Den kan me òg kalla «i110», og ligg der ein underkatalog der, heiter den kanskje både «i110/obl1» og «/Home/sstud2/t04/i110/obl1».

I kortnamn treng me ofte å klatra nedover mot rota. Då bruker me namnet «..» på neste katalog, så «/Home/sstud2» er det same som «..» frå heimen til t04, og «/» er det same som «../..». Den katalogen me står i heiter like eins «.», og det kjem me også til å bruka.

Me kan også bruka nokre forkortingar. «~» tyder heim, og «~manual» tyder heim til brukaren manual.

## 2.2.3 Innhald i katalogen (ls)

### Syntaks

```
ls [-F] [-l] [-d] [filnamn/katalognamn]
```

### Omtale

Sjølvsagt er der meir enn berre katalogar i filsystemet. Dokumenta og programma vert lagra i filer. Katalogane bruker me for å sortera filene slik at me lettare kan finna fram.

Me treng sjølvsagt å sjå kva som ligg i katalogen vår.

```
t04@bille 6 % ls
Mail      eksempelfila_mi  i110          tmp
t04@bille 7 %
```

Me har altså fire filer *eller* katalogar, men me kan ikkje sjå kva som er fil og kva som er katalog. «-F» legg til ein skrâstrek etter katalognamna, og «-l» gjev «lang listing».

```

t04@bille 7 % ls -F
Mail/          eksempelfila_mi  i110/          tmp/
t04@bille 8 % ls -l
total 8
drwx-----  2 t04      sstud          512 Jul 29 12:05 Mail
-rw-r--r--   1 t04      sstud          947 Jul 29 12:07 eksempelfila_mi
drwxr-xr-x   2 t04      sstud          512 Jul 29 12:06 i110
drwx-----  2 t04      sstud          512 Jul 29 12:05 tmp
t04@bille 9 %

```

No ser me at t04 har ein fil og tre underkatalogar i heimen sin. «d»en lengst til venstre av den lange lista står for «directory» som tyder katalog, så sjølv om me kan bruka «-F» og «-l» saman, så er det ikkje naudsynt. Den lange lista fortel oss òg kven som eig filene (t04) og kva gruppe som eig dei (sstud). Den fortel oss kor store dei er, fila er 947 teikn. Storleiken til katalogane bryr me oss sjeldan om. Datoen i lista fortel oss når nokon sist forandra fila. Me skal sjå meir på den lange lista seinare.

Me kan beda `ls` om å berre lista nokre filer eller katalogar. Bed me om at han skal lista ei fil, for me informasjon om den fila. Når me bed om ein katalog for me innhaldet. Me kan også få berre informasjon om ein fil, om me bruker opsjonen «-d».

```

t04@bille 9 % ls eksempelfila_mi
eksempefil_mi
t04@bille 10 % ls -l eksempelfila_mi
-rw-r--r--   1 t04      sstud          947 Jul 29 12:07 eksempelfila_mi
t04@bille 11 % ls -l Mail
total 6
-rw-r--r--   1 t04      sstud          1878 Jul 29 12:28 in
-rw-r--r--   1 t04      sstud           118 Jul 29 12:28 out
t04@bille 12 % ls -ld Mail
drwx-----  2 t04      sstud          512 Jul 29 12:28 Mail/
t04@bille 13 % ls -l -d Mail
drwx-----  2 t04      sstud          512 Jul 29 12:28 Mail
t04@bille 14 % ls -ld Mail eksempelfila_mi
drwx-----  2 t04      sstud          512 Jul 29 12:28 Mail
-rw-r--r--   1 t04      sstud          947 Jul 29 12:07 eksempelfila_mi
t04@bille 15 %

```

Du kan òg prøva

```

t04@bille 15 % ls /
t04@bille 16 % ls ..

```

## 2.2.4 Flytta seg i filsystemet (cd)

### Syntaks

`cd katalog`

### Omtale

Me ynskjer å kunna bruka så enkle kortnamn som mogeleg, difor flyttar me oss ofte omkring i filsystemet.

`cd` («change directory») utan argument bring oss heim, same kvar me er. Gjev me eit argument bring `cd` oss dit me bed om.

## Døme

```
t04@bille 17 % cd ..
t04@bille 18 % pwd
/Home/sstud2
t04@bille 19 % cd /tmp
t04@bille 20 % pwd
/tmp
t04@bille 21 % cd
t04@bille 22 % pwd
/Home/sstud2/t04
t04@bille 23 %
```

## 2.3 Filkommandoar

### 2.3.1 Nye katalogar (mkdir, rmdir)

#### Syntaks

```
mkdir katalog
rmdir katalog
```

#### Omtale

Me kan sjølv laga nye katalogar med `mkdir` («make directory»), og me kan fjerna gamle, tomme katalogar med `rmdir` («remove directory»).

#### Døme

```
t04@bille 23 % ls
Mail      eksempelfila_mi  i110          tmp
t04@bille 24 % mkdir test
t04@bille 25 % ls
Mail      i110            tmp
eksempelfila_mi  test
t04@bille 26 % rmdir test
t04@bille 27 % rmdir i110
rmdir: directory "i110": Directory not empty
t04@bille 28 % ls i110
obl1
t04@bille 29 % rmdir i110/obl1
t04@bille 30 % rmdir i110
t04@bille 31 % ls
Mail      eksempelfila_mi  tmp
t04@bille 32 %
```

Me kunne ikkje fjerna «i110» med ein gong, fordi han ikkje var tom, men det gjekk bra når me hadde fjerna «i110/obl1».

### 2.3.2 Flytta filer (mv, rm, cp)

#### Syntaks

```
cp [-i] gamalt_namn nytt_namn
cp [-i] fil [fil.] katalog
mv [-i] gamalt_namn nytt_namn
mv [-i] fil [fil.] katalog
```

```
rm [-r] [-f] [-i] fil [fil..]
```

## Omtale

`cp` («copy») kopierer filer, enter tar han ein fil og lagar ein kopi med nytt namn, eller så tek han ei liste med filer og kopierer dei til ein katalog.

`mv` («move») liknar, men han berre flyttar filene.

`rm` («remove») slettar filer. «`rm -r`» («recursive») kan også sletta katalogar med innhald. «`-f`» («force») fortel `rm` at du er heilt sikker og at han aldri skal spørja om lov til å sletta spesielle filar.

Om me bruker «`-i`» spør alle kommandoane om lov for kvar fil dei skal sletta eller skriva over.

## Døme

```
t04@bille 32 % cp ~manual/eksempelfila_mi .
t04@bille 33 % mkdir test
t04@bille 34 % mv eksempelfila_mi test/
t04@bille 35 % cd test/
t04@bille 36 % cp eksempelfila_mi ny_fil
t04@bille 37 % rm -i eksempelfila_mi ny_fil
rm: remove eksempelfila_mi (yes/no)? n
rm: remove ny_fil (yes/no)? y
t04@bille 38 %
```

Filene i heimekatalogen til `manual` (`~manual`) kan du gjerne kopiera for å øva deg, det er difor dei ligg der. Elles skal ein alltid spørja om lov før ein kopierer filene åt andre.

### 2.3.3 Visa filer (`less`, `cat`)

#### Syntaks

```
less [fil..]
cat [fil..]
```

## Omtale

`cat` listar ut innhaldet av dei filene du gjev. Ein lyt vera varsam så ein ikkje listar lange filer, for sjølv om det går for raskt til å lesa, så kan det vera lang tid å venta.

`less` listar filene side for side. Me kan blada opp og ned i fila med `k` og `j` eller med piltastane. Me kan raskt blada ei side med mellomromstasten («space»), og me kan søkja etter ord og uttrykk<sup>1</sup> med skråstrek. `less` viser søkjeuttrykket nedst på skjermen og du kan redigera det med slettetastar og piltastar. Når det er riktig trykkjer du vognretur.

Når me bed om fleire filer, brukar me «`:n`» for å få neste. For å avslutta tastar me `q` («quit»).

---

<sup>1</sup>Desse uttrykkja er såkalla regulære uttrykk, `regex(5)`. Når du har lært om kommandoen `man` kan du bruka han for å læra meir om `regex(5)`.

## Døme

```
t04@bille 97 % cat kortfil
Dette er berre ei kort fil for å kunna øva seg.
t04@bille 98 % less help.txt
```

## 2.3.4 Utskrift (lpr, lprm, lpq, enscript)

### Syntaks

```
lpr [-Plp8] [fil..]
lpq [-Plp8]
lprm [-Plp8] n
lprm [-Plp8] -
```

### Omtale

lpr sendar ei eller fleire filer til skrivaren. Om me ynskjer å senda filene til ein særskild skrivar, brukar me opsjonen «-P». Skrivaren på UA heiter «lp8». Skrivaren på FakLabben heiter «lp9». Kontoane skal vera sette opp slik at utskriftene kjem på riktig skrivar i utgangspunktet.

lpq visar oss kva filer som ligg i køen til skrivaren og kven som har sent dei. Kvar fil har eit nummer, eit «job»-nummer.

lprm trekkjer fil nr. *n* tilbake frå køen. Skriv med ein minus (-) i staden for eit nummer, trekkjer me alle våre filer tilbake.

## Døme

```
t04@bille 22 % lpr eksempelfila_mi
t04@bille 23 % lpq
lp8 is ready and printing
Rank  Owner  Job   File(s)                Total Size
active s689   975   standard input         3060601 bytes
1st   ip387   694   /var/tmp/aaa0IbhI1     722 bytes
2nd   s878    344   standard input         1467443 bytes
3rd   ip357   144   standard input         14715 bytes
4th   s878    345   standard input         23999 bytes
5th   t04     573   eksempelfila_mi       947 bytes
t04@bille 26 % lprm 573
lp8-573: cancelled
t04@bille 27 % lpq
lp8 is ready and printing
Rank  Owner  Job   File(s)                Total Size
active s689   975   standard input         3060601 bytes
1st   ip387   694   /var/tmp/aaa0IbhI1     722 bytes
2nd   s878    344   standard input         1467443 bytes
3rd   ip357   144   standard input         14715 bytes
4th   s878    345   standard input         23999 bytes
t04@bille 28 % lpr eksempelfila_mi
t04@bille 30 % lpr kortfil
t04@bille 31 % lpq
lp8 is ready and printing
Rank  Owner  Job   File(s)                Total Size
active s689   975   standard input         3060601 bytes
1st   ip387   694   /var/tmp/aaa0IbhI1     722 bytes
2nd   s878    344   standard input         1467443 bytes
3rd   ip357   144   standard input         14715 bytes
4th   s878    345   standard input         23999 bytes
```

```

5th    t04    574    eksempelfila_mi    995 bytes
6th    t04    575    kortfil            48 bytes
t04@bille 32 % lprm -
lp8-574: cancelled
lp8-575: cancelled
t04@bille 33 % lpq
lp8 is ready and printing
Rank   Owner   Job    File(s)                Total Size
active s689   975    standard input         3060601 bytes
1st    ip387   694    /var/tmp/aaa0IbhI1    722 bytes
2nd    s878   344    standard input         1467443 bytes
3rd    ip357   144    standard input         14715 bytes
4th    s878   345    standard input         23999 bytes
t04@bille 34 %
\end{verbatim}

```

### 2.3.5 Avbryta eit program (Ctrl-C)

Me kan avbryta dei aller fleste program og kommandoar med Ctrl-C. Om du skulla vera uheldig å skriva feil og starta noko som tar lang tid, kan du alltid trykkja Ctrl-C. Eg skal skriva meir om dette i kapittel 6.

```

t04@bille 37 % ls -l /Home/sstud2
^C

```

## 2.4 Tilgangar

Dei fleste vil halda nokre av filene sine hemmelege, medan dei treng å dela nokre med andre. Svært sjeldan ynskjer me at andre skal kunna endra våre eigne filer. Difor kan me fastsetja kva filer andre skal kunna lesa og kva andre skal få skriva til.

Lat oss sjå på den lange lista fra `ls` igjen.

```

t04@bille 8 % ls -l
total 8
drwx----- 2 t04    sstud    512 Jul 29 12:05 Mail
-rw-r--r--  1 t04    sstud    947 Jul 29 12:07 eksempelfila_mi
drwxr-xr-x  2 t04    sstud    512 Jul 29 12:06 i110
drwx----- 2 t04    sstud    512 Jul 29 12:05 tmp

```

Tilgangane er markert ved teikna 2-10 på kvar line. Der er ni teikn, tre grupper med tre teikn i kvar gruppe. Den fyrste teikngruppa er for eigaren sjølv, t04, den andre er for gruppa, sstud, og den tredje er for alle dei andre.

Det fyrste teiknet i kvar gruppe er anten «r» for «read» for å gjeva rett til å lesa fila, eller «-» dersom der ikkje er lesetilgang. Det andre teiknet er anten «w» for «write» for skrivetilgang, eller «-». Det siste er «x» («execute») dersom der er tilgang til å køyra fila som eit program eller opna katalogen dersom det er ein katalog.

Me kan sjå at den einaste som kan endra filer her er brukaren sjølv. eksempelfila\_mi er ikkje eit program, så ingen kan køyra ho, men dei kan lesa ho. i110 kan også alle lesa og opna, t04 vil har posten sin i katalogen Mail (sjølv sagt) og katalogen tmp heilt for seg sjølv.

Det kan vera vanskeleg å forstå korleis ein nyttar «r» og «x» på katalogar. I regelen kan ein seia at anten gjev ein bae, eller så gjev man ingen av delane. For å kunna sjå kva filer som ligg i katalogen, må man ha «r», og for å gjera

noko me dei må man ha «x» på katalogen, i tillegg til tilgang til den enkelte fil eller underkatalog.

Men kan sjå nokre fleire variantar:

```
t04@bille 1 % ls -l ~manual
total 46
-rw-rw-r-- 1 manual manual      947 Jul 29 16:54 eksempelfila_mi
-rw-rw-r-- 1 manual manual     4766 Jul 29 17:21 help.txt
-rw-rw-r-- 1 manual manual      48 Jul 29 17:18 kortfil
lrwxrwxrwx 1 manual manual      31 Jul 28 19:34 manual -> /net/www/ii/fu/M
anualer/Handbok
drwxrwsr-x 3 manual manual     8192 Aug 11 16:34 public_html
drwxrws--- 4 manual manual     8192 Aug 11 16:30 work
t04@bille 2 % ls -ld /tmp
drwxrwxrwt 9 sys      sys      575 Aug 11 20:00 /tmp
t04@bille 3 %
```

I heimekatalogen til manual, har gruppa og brukaren dei same rettane. Gruppa manual er me som skriv denne handboka, og alle i gruppa skal kunne gjera endringer i eksempelfilene og manualen. Arbeidskatalogen «work» er stengd for andre.

«/tmp» (forkorting for «temporary») er ein felleskatalog, kor alle brukarar kan skriva og lesa. Det tiande teiknet er ein «t» i staden for ein «x», og det tydar at sjølv om alle og einkvar kan skriva i katalogen, så kan dei ikkje utan vidare sletta filar som dei ikkje eig sjølv.

Det sjuande teiknet er ein «s» i staden for ein «x» og betyr at gruppa (manual) som eig katalogen, òg skal eiga alle dei nye filene som vert lagra der.

## 2.4.1 Endra tilgangar (chmod)

### Syntaks

```
chmod [-R] tilgang fil [fil..]
```

### Omtale

chmod («change mode») endrar tilgangane til ein eller fleire filer.

Dei ni teikna i lista frå ls er ni brytarar. Kvar brytar kan vera av eller på. I lista frå ls tydar «-» at han er, medan bokstaven tydar at han er på.

Den enklaste måten å bruka chmod er å seia kva teikngrupper me skal endra; om me skal slå brytarar på eller av; og til sist kva brytarar som skal endrast.

Teikngruppene er

u	user	brukaren
g	group	gruppa
o	other	andre
a	all	alle

Endringane er:

+	slå på brytarar
-	slå av brytarar
=	sett brytarar

Og dei tre brytarane er som i lista frå ls:

r	read	lesa
w	write	skriva
x	execute	opna eller køyra

Til dømes har me «og-w» for å stengja leseretten for alle andre enn brukaren sjølv. «r»- og «x»-brytarane vert ikkje råka. «u+rw» sikrar alle rettar for brukaren. «o=rx» slår skriveretten av og opne- og leserettane på for andre brukarar.

Opsjonen «-R» («recursive») tydar at `chmod` skal gå gjennom katalogar og underkatalogar og gjera den same endringa med alle filene han finn.

### Døme

```
t04@bille 10 % chmod a-w eksempelfila_mi
t04@bille 11 % chmod a+x hello.sh
t04@bille 12 % ls -l
total 22
drwx----- 2 t04      sstud      512 Jul 31 23:56 Mail
-r--r--r--  1 t04      sstud      947 Aug  9 16:11 eksempelfila_mi
-rwxr-xr-x  1 t04      sstud        29 Aug 11 20:25 hello.sh
-rw-r--r--  1 t04      sstud    4766 Jul 29 17:21 help.txt
-rw-r--r--  1 t04      sstud        48 Jul 29 17:18 kortfil
drwxr-xr-x  2 t04      sstud      512 Jul 29 17:04 test
drwx----- 2 t04      sstud      512 Jul 29 12:05 tmp
t04@bille 13 %
```

# Kapittel 3

## Rettleiinger

### 3.1 Unix-manual på maskina

#### 3.1.1 man, apropos, whatis

##### Syntaks

```
man [-s n] [-M sti] [-a] [-l] kommando
whatis kommando
apropos nøkkelord
```

##### Beskriving

Unix vert levert med ei handbok som me kan lesa direkte på maskina. Til dette bruker me programma `man`, `whatis` og `apropos`. `man` viser heile man-sida for ein kommando. `man` bruker programmet `less`<sup>1</sup>, så du kan bruke dei same kommandoane for å flytta omkring og for å søka etter bestemte ord. Sjølvstund brukar du òg `q` for å avslutta.

Prøv til dømes

```
t04@bille 2 % man man
Reformatting page. Wait... done
[...]
```

`whatis` gir kun ei kort beskriving på ei line. Det er lina som står under «NAME» i man-sida.

`apropos` søker gjennom alle man-sidane og listar dei som passar med nøkkelordet.

Man-sidane er inndelt i kapitlar:

- 1 allminnelege brukerkommandoer
  - 2-3 funksjoner og systemkall for programmering
  - 4 filformater
  - 5 ymse
  - 6 spel og demoar
  - 7-9 administrative program og filer
- Mange av kapitla har

underkapitlar markert ved ein bokstav. Til dømes er dei administrative kom-

---

<sup>1</sup>Vel, det er det vanlege, men du kan egentlig velga eit annet program sjølv.

mandoane som kun ingeniørane treng, samla i avsnitt 1m. Ynskjer du å slå opp i eit bestemt avsnitt, kan du gje `man` opsjonen «-s».

Dei fleste brukarar les mest berre kapittel 1 og fra tid til anna spesielle ting i kapitla 4 og 5. Me som programmerer C og Fortran, treng også kapitla 2 og 3.

Når ein skriv om ein kommando er det vanleg å gje kapittelet og evt. underkappitelet til `man`-sida òg, til dømes skriv eg gjerne om `man(1)`, for å fortella at folk kan lesa meir i om `man` man-sidene, kapittel 1.

Mange kommandoer har fleire man-sider, enten i ulike kapittel eller i ulike kataloger i filsystemet. Du kan gi en bestemt man-katalog med «-M», se under «Døme» nedenunder.

«`man -l`» gir en liste over man-sider for kommandoen, medan «`man -a`» viser alle man-sidane. `less` viser da ei side av gangen og du taster «:n» for å koma til neste.

## Døme

```
t04@bille 28 % apropos editor
ed          ed (1)          - text editor
edit       edit (1)       - text editor (variant of ex for casual users)
ex         ex (1)         - text editor
[...]
vedit      vi (1)         - screen-oriented (visual) display editor basedo
n ex
vi         vi (1)         - screen-oriented (visual) display editor basedo
n ex
view      vi (1)         - screen-oriented (visual) display editor basedo
n ex
[...]
t04@bille 29 % whatis ls
ls         ls (1)         - list contents of directory
ls         ls (1b)        - list the contents of a directory
[...]
t04@bille 30 % man -l ls
ls (1)    -M /usr/man
ls (1b)   -M /usr/man
t04@bille 31 % man -s 1b ls
Reformatting page.  Wait... done
[...]
```

`apropos` og `whatis` gir tre kolonner. Den fyrste er kommandonamnet, den andre er sidenamnet og den tredje er beskrivinga. Legg merke til at `vedit`, `vi` og `view` deler ei side.

«`man -l`» gir i tillegg til kapitla også man-katalogen, som du kan oppgi med «-M».

Til slutt slår me opp i kapittel 1b for å lesa om `/usr/ucb/ls`, som er litt ulik `/usr/bin/ls` som me ellers bruker.

# Kapittel 4

## Dei fyrste programma

### 4.1 Redigeringsprogram

#### 4.1.1 vi, vim

##### Syntaks

```
vi [filnamn]
vi -R [filnamn]
vim [filnamn]
vim -R [filnamn]
```

##### Omtale

vi er det gode gamle, klassiske tekstredigeringsprogrammet for Unix. Du kan rekna med å finna han på dei aller fleste unixmaskiner. Trass i at ein lang rekkje med kraftige funksjonar, er han også svært rask. vim er «*vi improved*», og i hovudsak verkar dei to likt. vim kan i tillegg mykje meir, men som vi får sjå er der berre nokre få skilnader vi har tid til å sjå på. Ingeniørane ved instituttet støttar vi, men ikkje vim.

Namnet, «vi», kjem frå «visual»; han er eit *visuelt* redigeringsprogram. Dette kan verka litt rart i våre dagar, for du har sikkert ikkje sett noko redigeringsprogram som ikkje er visuelt. Me har òg ikkje-visuelle program her, dei heiter *ed* og *ex*. Den siste er forresten berre ein variant av vi.

##### Me startar og avbryt

Me startar vi på ein fil:

```
t04@bille 5 % vi ei_ny_fil
~
~
~
~
~
"eksempelfila_mi" 0 lines, 0 characters
```

Linene med tilda (~) finst ikkje i fila, dei er der for å fylla skjermen når fila er tom. Den nedste lina fortel kva fil du arbeid med og kor stor ho er.

Kan henda merk du med ein gong at du ikkje får skriva noko. Det er fordi vi startar i kommandotilstand og alt det du skriv, oppfattar maskina som kommandoar. Kommandoane for å avslutta kan me like godt læra med det same:

```
:q          avslutta
:q!         avslutta utan omsyn til fila
:w [fil]   lagra på fil
:w! fil    lagra på fil utan omsyn til hennar gamle innhald
:wq        lagra og avslutta
```

Alle desse er *kolonkommandoar* (eller *ex-kommandoar*). Når du tastar kolon (: ) ser du at den nedste lina forandrar seg og du kan lesa kva du skriv. Om du skriv noko feil kan du sletta bakover eller du kan avbryta kommandoen med Ctrl-C. I vim kan du òg bruka piltastane. Når du har skrive heile kommandoen slik han skal vera, tastar du vognretur eller escape.

Den fyrste kommandoen «:q» prøver berre å avslutta, men om du har endra noko i fila, så vil han ikkje. Den andre kommandoen «:q!» avsluttar same kva, utan å gjera noko med fila i filsystemet. Då mistar du alt du har gjort sidan sist du lagra fila.

For å lagra fila bruker du «:w». Om du gjev eit filnamn som parameter lagrar du fila under det nye namnet. Om der allereie er ei fil med det nye namnet, vil ikkje vi overskriva ho utan at du skriv «:w! fil».

Om du vil lagra fila og avslutta med det same, kan du gjera det med «:wq».

## Når noko går galt

Om du byrjar å skriva noko feil, eller kjem ut av kommandotilstanden ved eit uhell, skal du alltid hugsa «escape»-tasten (ESC). Når du trykkar «escape» avbryt vi det han gjør, og ventar på ein ny kommando. I innsettjingsstilstanden bring «escape» oss tilbake til kommandotilstanden. Om du har skrive ein halv kommando, så gløymer vi det når du trykkar «escape».

Du kan også gjera om den siste kommandoen med u («undo»). I vi kan du gå eit steg tilbake. Trykkjer du u ein gong til, så er det u-kommandoen du tek tilbake.

I vim kan du gå mange steg tilbake ved å tasta u fleire gongar. Om du går får langt tilbake, så går du fremat med Ctrl-R.

## Flytta omkring

Lat oss no starta vi på ei gamal fil:

```
t040bille 6 % vi eksempelfila_var
```

Me kan flytta oss (dvs. flytta markøren) omkring med tastane:

```
h  høgre
j  ned
k  opp
l  venstre
```

Nokre stadar kan me òg bruka piltastane. Det lyt ein prøva.

Me kan hoppa omkring på lina:

0	byrjinga av lina
^	byrjinga av lina, etter blankteikn
b	beginning byrjinga av ordet
e	end slutten av ordet
\$	slutten av line

Desse kommandoane lat oss blada raskt i fila:

Ctrl-B	back	ei skjermleingd opp
Ctrl-U	up	ei halv skjermleingd opp
Ctrl-D	down	ei halv skjermleingd ned
Ctrl-F	forward	ei skjermleingd ned

Me kan raskt gå til line er. # med «#G» eller «:#».

**Gjentaka kommandoar** Me kan gjentaka ein kvar vi-kommandoar (ikkje kolonkommandoane) ved å tasta talet på gjentak rett før kommandoen. Til dømes:

15b	går 15 ord tilbake
15k	går ned 15 liner
6Ctrl-U	rullar seks halve skjermbilete opp

**Søkja** Me kan søkja etter ord (eig. regulære uttrykk, men det kan du lesa om i man-sida [regex(5)]) i fila med skråstrek (/). Når du tastar skråstrek, kjem søkjeuttrykket opp nedst på skjerna. Prøv sjølv.

Me kan raskt byta ut enkelte uttrykk i heile fila, med

```
:%s/gamalt/nytt/opsjonar
```

Kommandoen er eit døme på ein godt samansett kolonkommando. Han bytter ut *gamalt* med *nytt*. Der er eit par vanlege opsjonar:

c	confirm	spørj om lov for kvart byte
g	global	byt ut alle førekomstar, ikkje berre den fyrste på kvar line
p	print	skriv ut lina efter bytet

```
:%s/hans/Hans/g
:%s/ / /g
:%s/;/;/c
:%s/;[^;]*$/;/
:%s/\(.*\):\(.*\)/\2:\1/
```

Den nest siste kommandoe slettar alt som komer etter det siste semikolonet på kvar line. Om du skal forstå den siste kommandoen, så får du lesa ein annan stad...

## Leggja til tekst

Der er ein rekkje med kommandoar for å leggja inn ny tekst:

o	open line	opna ny line under denne
O	Open line	opna ny line over denne
i	insert	sette inn teikn før markøren
I	Insert	sette inn teikn i byrjinga av lina
a	append	sette inn teikn etter markøren
A	Append	leggja til teikn på slutten av lina

Alle desse bring oss til innsetjingstilstand. Når me har innsetjingstilstand, vert alt me skriv lagt til i teksten, og me kan bruka slettetasten. I vim kan ein òg flytta markøren med piltastane, om maskina har piltastar som verkar.

## Endra fila

For å byte ut teikn har me to kommandoar:

```
r  replace  erstatt eit teikn
R  Replace  erstatt fleire teikn
```

berre den siste av dei bring oss til innsetjingstilstand.

Me har to kommandoar for å sletta enkeltteikn:

```
x  slett teiknet under markøren
X  slett bakover
```

Til slutt har me to kommandoar som me brukar saman med ein kommando for å flytta markøren. Om ein tastar *d* (for «*delete*») og ein kommando for å flytta seg, vil all tekst me flyttar forbi forsvinna. *c* (for «*change*») verkar på same vis, det vert det same som å fyrst bruka *d* og så tasta *i*. Dvs. me fjernar teksta me flyttar forbi og sett samstundes inn ny. «*dd*» og «*cc*» slettar eller bytar ut ei heil line.

```
cw  change word  byt ut ord
c$  change word  byt ut resten av lina
c4l change word  byt ut fire teikn
dd  delete line  slett lina
dG  delete line  slett resten av fila
5dd delete line  slett fem liner
```

## Hjelp i vim

I vim kan du lesa hjelpefiler med kommandoen «:he» eller «:help». Du kan få generell hjelp, eller du kan spørja om ein særskild kommando:

```
:help
:help help
:he substitute
```

## Fleire kilder

Eg har mi eiga side om vi kor du kan finna fleire og større manualar. Du finn ho på <http://www.ii.uib.no/%7egeorg/doc/vi/>.

# Kapittel 5

## Internet

### 5.1 Internet Generelt

#### 5.1.1 Kva tenester finns på Internet

**WWW** er tekstinformasjon på internett, der ein kan legge inn bilete, og følge såkalla hyperlinker til andre sider på nettet. Dei siste åra har det blitt vanleg å overføre musikk, film og anna via www.

**Epost** (Elektronisk post) er å sende ei melding elektronisk til ein annan brukar på Internet. Er kjend for å vere raskare enn vanleg post

**Usenet News** - Diskusjonsgrupper. Det er fleire tusen av desse, og det finns grupper for nesten ei kvar interesse. Det er både seriøse og mindre seriøse grupper.

**FTP** er overføring av filer frå ein stad til ein annan. Det er enten frå ein annan stad du kan logge inn, eller anonym ftp, der du kan hente program og data frå ftpbasar.

Det finns utanom dette ei rekke andre meir eller mindre spesialiserte tenester på internett. Dette kan vere alt frå overføring av betaling til pratekanalar, som IRC. I tillegg er telnet, rlogin, finger og andre tilsvarande kommandoar sjølvsagt og tenester på Internet.

#### 5.1.2 Epost generelt

Epost er å sende ei tekstmelding til ein annan brukar på Internet. Du kan og sende andre typar data, men i utgangspunktet er det tekst du sender. Han du skal sende til må då sjølvsagt ha ei epostadresse.

#### 5.1.3 Epostadressar

Ei postadresse på internett er bygd opp av to delar, brukarnamn og domene-namn, skild av ein krøllalfa. Skal du til dømes sende epost til ein med brukar-namn 's007' her på instituttet, vil epostadressa bli  
s007@lstud.ii.uib.no.

's007' er brukarnamnet, og domenenamnet er 'ii.uib.no', som er domenet til institutt for informatikk ved universitetet i Bergen. Dersom du er logga på ei av maskinene til instituttet, og skal sende til ein annan på instituttet, treng du ikkje å ha med «@» og domenenamn. Postsystemet vil da gå ut frå at du sender til ein brukar på på same stad.

#### 5.1.4 Hovud og kropp

#### 5.1.5 dtmail, funksjonatitet, sende og ta imot epost

Eit epostprogram som er enkelt å bruke er dtmail. dtmail kan du starte opp på verktøylinja nederst på skjermen, eller med kommandoen 'dtmail' frå kommandolinja. Da startar dtmail opp, og gir deg ein oversikt over kva meldingar du har i postboksen.

/snapshot av dtmail/

#### 5.1.6 Lese epost

dtmail vil til vanleg vise den første uleste postmeldinga i det nederste vindauget. Du kan trykke på 'next' for å lese den neste uleste meldinga, eller på 'previous' for å lese tidlegare meldingar. Du kan og trykke på ein vilkårleg melding, og da vil denne meldinga kome opp i sit eige vindauge.

#### Sende epost

Når du har fått ei epostmelding, kan du trykke på 'reply to sender', for å svare på meldinga. Da vil du få opp eit vindauge der du kan skrive inn eit svar på meldinga. Over dette feltet er det eit 'To'-felt der epostadressa til den du skal sende meldinga til står, eit 'Subject'-felt der ein skriv inn kva temaet for epostmeldinga er, og eit 'CC'-felt, der ein skriv inn epostadressa til folk som skal få kopi av svaret.

Dersom du skal sende epost som ikkje er svar på ei melding, velger du 'New Message' på compose-menyen. Du vil få opp eit vindauge akkurat som når du svarar på ei melding, men adressefeltet og tittelfeltet vil vere tomt. Du må no fylle inn med epostadresse og tittel sjølv.

#### 5.1.7 Filer som vedlegg(attachements)

I tillegg til tekst kan du ta med filer av ulikt slag som vedlegg. Dette gjer du ved å velge 'add file' frå attachment-menyen i vindauget du skriv inn epost. Da får du opp ein boks, der du kan skrive inn namn på fila du vil sende. Du kan sende fleire slike filer i kvar epostmelding. Dette kan vere ein praktisk måte å sende filer til andre brukarar på Internet.

Sjølv om dei fleste kan lese vedlegg til epost, er det framleis nokon som ikkje kan lese vedlegg.

Du bør heller ikkje sende filer frå tekstbehandlingsprogram som vedlegg, dersom du sende ein tekst. Det kan ofte gi problem for dei som les i andre enden, dersom du sender meldinga på denne måten. Det er ikkje alltid dei i andre enden har programmet som skal til for å lese fila. Det same gjeld nokre postlesarar som sender HTML (www-tekst) i epostmeldinga. Desse inneholder

kontrollkoder som gjer dei vanskeleg å lese utanom ein www-lesar. Vanleg tekst skal derfor sendast som tekst.

### 5.1.8 mailx/UCB mail

#### Syntaks

```
mailx [-f +mappe/fil]
mailx [-s emne] adresse
```

#### Omtale

mailx(1) liknar den gode gamle postlesaren frå BSD unix (UCB). Fordi han er enkel, er han òg rask, og for dei som har lært å bruka han er han betre en nokon annan me har sedd. mailx kan og eit par ting som dei andre postlesarane ikkje kan, til dømes å ta inndata frå ei vilkårleg fil.

Uten argument åpnar mailx mappen med den nye posten din. Skal du lesa posten i ein mappe du har laga tidlegare, seier du «mailx -f +<mappe>», og skulle du ønskja å lesa post i ein eller anna fil, kan du seia «mailx -f <filnamn>».

Om du vil senda post skriv du berre «mailx adresse». Du kan skriva emnefeltet på kommandolina med opsjonen «-s», men du kan òg skriva det etter at mailx er starta.

#### Les post

Når me skal lesa ny post skriv me berre mailx:

```
t04@bille 4 % mailx
mailx version 5.0 Tue Jul 15 21:29:48 PDT 1997  Type ? for help.
"/var/spool/mail/t04": 2 messages 1 new 2 unread
  U  1 Hans Georg Schaath Sun Aug 16 03:15  22/795  test
 >N  2 Hans Georg Schaath Sun Aug 16 03:18  18/765  Ny test
Mail >
```

og mailx tek over og gjev oss sitt eiga klarteikn, «Mail >».

mailx fortel oss ein masse med ein gong. Mellom anna kan ein bruka spørsmålsteiknet for å få hjelp.

Han les fila /var/spool/mail/t04/var/spool/mail/t04 og finn to meldingar. Eg har aldri lese nokon av dei, men berre den eine er ny. Begge meldingane kjem frå meg sjølv. Dei er sende Søndag 16. august i tretida om morgonen. Den fyrste er 22 liner og 795 teikn, medan den andre er 18 liner og 765 teikn. Titlene er «test» og «Ny test». Heilt til venstre markere vinkelen (>) den aktive meldinga. Bokstavane fortel om eg har lese meldinga.

```
U  unread  ikkje lest
R  read    lest
O  old     lese tidligare
```

For lesa den aktive meldinga trykkjer eg vognretur. mailx brukar less(1) for å visa ho. Eg kan lesa ei særskild melding ved å skriva nummeret hennar og trykkja retur. Når eg har lese den aktive meldinga, bring vognreturen meg til den neste meldinga.

Dei viktigaste kommandoane er desse:

s [ <i>nr</i> ]	mappe	save	lagra og sletta melding <i>n</i> i <i>mappe</i>
c [ <i>nr</i> ]	mappe	copy	lagra melding <i>n</i> i <i>mappe</i>
d [ <i>n</i> ]		delete	sletta melding <i>n</i>
r [ <i>n</i> ]		reply	svara
R [ <i>n</i> ]		reply all	svara til sendaren og alle mottakarane
m [ <i>adresse</i> ]		mail	skriva melding til <i>adresse</i>
h [ <i>n</i> ]		headers	visa hoder
z [+/-]		(headers)	visa neste/forrige side med hoder
folder	<i>mappe</i>	folder	opna <i>mappe</i>
q		quit	avslutta og skriva endringar til postmappen
x		exit	avslutta utan å skriva endringar til postmappen

Meldingsnumra *n* er valfrie. Når eg ikkje gjev noko nummer, får eg den aktive meldinga. Postmappene skriv eg i regelen med ein pluss (+) framfør. Då brukar `mailx` ei fil i postkatalogen `min (/Mail/)`.

Ein lyt vera varsom og ikkje bruka den store «R»-en for mykkje. Mykkje post vert send til postlister som `ipstud` og `il10`, og det kjem ofte svar som berre var meint for sendaren, men som vert send til alle ved eit uhell.

### Skriva post

Det er tre måter å skriva meldingar på. Eg kan svara på ei melding med «r» eller «R». Eg kan laga ei ny melding med «m» når eg allereie køyrer `mailx`, eller eg kan starta `mailx` med ei adresse som argument.

Hvis adressa og/eller tittelen ikkje er gjeve (og inndataene ikkje kjem frå fil, jfr. avsnitt 6.2.), vil `mailx` spørja. Etterpå får eg ei blank linje, utan noko klarteikn, og `mailx` ventar på teksten til meldinga. Eg kan utan vidare byrja å skriva, og når eg er ferdig, avsluttar eg med punktum (.) på ei eiga line.

Tilda kan eg ikkje bruka utan vidare på byrjinga av lina. `mailx` brukar ho som eit kommandoteikn. Ønskjer eg éi tilde, skriv eg `to`. To kommandoar er særst nyttige:

```

ẽ          edit          startar eit redigeringsprogram
m̃ [n]    message       sitér ei melding

```

«ẽ» gjør det mogeleg å skriva meldinga i eit redigeringsprogram. Når programmet avsluttar kjem ein tilbake til `mailx`, og ein kan skriva ei line med berre punktum for å sende meldinga.

«m̃» kan berre brukast når ein har køyrd «m», «r» eller «R» frå `mailx`. Han tek inn melding *n* eller meldinga eg svarar på, som eit sitat. Etter «m̃» lyt ein starta eit redigeringsprogram med «ẽ» og sletta dei delane av sitatet ein ikkje treng. Det er ein uskikk å sitera meir enn naudsynt.

### 5.1.9 Andre postprogram på instituttet

`dtmail` er ein enkel postlesar. Mange meiner denne ikkje er avansert nok, eller må ha ein annan epostlesar som kan startast utanom CDE, eller vil ha andre eigenskapar som `dtmail` ikkje har. Det er fleire alternative postlesarar. Dersom du vil ha ein postlesar som ikkje køyrer i sitt eige vindauge, er `pine` og `mutt` bra alternativ. Andre ganger er det ein fordel å ha ein liten og enkel postlesar, som `mailx`.

## 5.1.10 mailx/UCB mail

### Syntaks

```
mailx [-f +mappe/fil]
mailx [-s emne] adresse
```

### Omtale

mailx(1) liknar den gode gamle postlesaren frå BSD unix (UCB). Fordi han er enkel, er han òg rask, og for dei som har lært å bruka han er han betre en nokon annan me har sedd. mailx kan og eit par ting som dei andre postlesarane ikkje kan, til dømes å ta inndata frå ei vilkårleg fil.

Uten argument åpnar mailx mappen med den nye posten din. Skal du lesa posten i ein mappe du har laga tidlegare, seier du «mailx -f +<mappe>», og skulle du ønskja å lesa post i ein eller anna fil, kan du seia «mailx -f <filnamn>».

Om du vil senda post skriv du berre «mailx adresse». Du kan skriva emnefeltet på kommandolina med opsjonen «-s», men du kan òg skriva det etter at mailx er starta.

### Les post

Når me skal lesa ny post skriv me berre mailx:

```
t04@bille 4 % mailx
mailx version 5.0 Tue Jul 15 21:29:48 PDT 1997  Type ? for help.
"/var/spool/mail/t04": 2 messages 1 new 2 unread
  U  1 Hans Georg Schaath Sun Aug 16 03:15   22/795   test
 >N  2 Hans Georg Schaath Sun Aug 16 03:18   18/765   Ny test
Mail >
```

og mailx tek over og gjev oss sitt eiga klarteikn, «Mail >».

mailx fortel oss ein masse med ein gong. Mellom anna kan ein bruka spørsmålsteiknet for å få hjelp.

Han les fila /var/spool/mail/t04/var/spool/mail/t04 og finn to meldingar. Eg har aldri lese nokon av dei, men berre den eine er ny. Begge meldingane kjem frå meg sjølv. Dei er sende Søndag 16. august i tretida om morgonen. Den fyrste er 22 liner og 795 teikn, medan den andre er 18 liner og 765 teikn. Titlene er «test» og «Ny test». Heilt til venstre markere vinkelen (>) den aktive meldinga. Bokstavane fortel om eg har lese meldinga.

```
U  unread   ikkje lest
R  read     lest
O  old      lese tidligare
```

For lesa den aktive meldinga trykkjer eg vognretur. mailx brukar less(1) for å visa ho. Eg kan lesa ei særskild melding ved å skriva nummeret hennar og trykkja retur. Når eg har lese den aktive meldinga, bring vognreturen meg til den neste meldinga.

Dei viktigaste kommandoane er desse:

s [ <i>nr</i> ]	mappe	save	lagra og sletta melding <i>n</i> i <i>mappe</i>
c [ <i>nr</i> ]	mappe	copy	lagra melding <i>n</i> i <i>mappe</i>
d [ <i>n</i> ]		delete	sletta melding <i>n</i>
r [ <i>n</i> ]		reply	svara
R [ <i>n</i> ]		reply all	svara til sendaren og alle mottakarane
m [ <i>adresse</i> ]		mail	skriva melding til <i>adresse</i>
h [ <i>n</i> ]		headers	visa hoder
z [ <i>+/-</i> ]		(headers)	visa neste/forrige side med hoder
folder	<i>mappe</i>	folder	opna <i>mappe</i>
q		quit	avslutta og skriva endringar til postmappen
x		exit	avslutta utan å skriva endringar til postmappen

Meldingsnumra *n* er valfrie. Når eg ikkje gjev noko nummer, får eg den aktive meldinga. Postmappene skriv eg i regelen med ein pluss (+) framfør. Då brukar `mailx` ei fil i postkatalogen min (`~Mail/`).

Ein lyt vera varsom og ikkje bruka den store «R»-en for mykkje. Mykkje post vert send til postlister som `ipstud` og `i110`, og det kjem ofte svar som berre var meint for sendaren, men som vert send til alle ved eit uhell.

### Skriva post

Det er tre måter å skriva meldingar på. Eg kan svara på ei melding med «r» eller «R». Eg kan laga ei ny melding med «m» når eg allereie køyrer `mailx`, eller eg kan starta `mailx` med ei adresse som argument.

Hvis adressa og/eller tittelen ikkje er gjeve (og inndataene ikkje kjem frå fil, jfr. avsnitt 6.2.), vil `mailx` spørja. Etterpå får eg ei blank linje, utan noko klarteikn, og `mailx` ventar på teksten til meldinga. Eg kan utan vidare byrja å skriva, og når eg er ferdig, avsluttar eg med punktum (.) på ei eiga line.

Tilda kan eg ikkje bruka utan vidare på byrjinga av lina. `mailx` brukar ho som eit kommandoteikn. Ønskjer eg éi tilde, skriv eg to. To kommandoar er særst nyttige:

```
ẽ edit          startar eit redigeringsprogram
m̃ [n] message  sitér ei melding
```

«ẽ» gjør det mogeleg å skriva meldinga i eit redigeringsprogram. Når programmet avsluttar kjem ein tilbake til `mailx`, og ein kan skriva ei line med berre punktum for å sende meldinga.

«m̃» kan berre brukast når ein har køyrd «m», «r» eller «R» frå `mailx`. Han tek inn melding *n* eller meldinga eg svarar på, som eit sitat. Etter «m̃» lyt ein starta eit redigeringsprogram med «ẽ» og sletta dei delane av sitatet ein ikkje treng. Det er ein uskikk å sitera meir enn naudsynt.

### 5.1.11 Postlister: kurslister, sstud, hstud, majordomo

I tillegg til å sende epost til ein person, kan ein sende post til ein postliste. Da vil alle som er med på postlista få epostmeldinga. Her på instituttet er det fleire epostlister for faglege og andre formål. Dersom du sender epost til 'i110', vil alle som tar i110 få epostmeldinga. På same måte finns det postlister for dei fleste av kursa.

terminal - Alle terminalvaktene trouble - Melding av tekniske feil på anlegget.  
fu - Alle som er med i fagutvalet pasta - Dersom du vil ha noen å spise pasta sammen med

ipstud - Alle på grunnkurs(i110, i160) sstud - Alle med s-konto (Dei fleste lågaregradstudentar) hstud - Alle hovudfagstudentar

Det finns fleire postlister, både her og utanfor instituttet for dei som er interessert i eit spesielt tema. Korleis ein kan melde seg på postlister, står det om i neste avsnitt.

Terminal kan du alltid sende epost til, dersom du lurar på noko. Har du faglege spørsmål, vil gruppeleiarane svare på spørsmål, men dersom du har noko som gjeld alle på kurset, kan du sende epost til kurslista. Nokre av kurslistene må ein melde seg på sjølv.

Studentlistene (sstud, hstud, ipstud) bør du ikkje sende til utan at du har noko veldig viktig å komme med. Ein god vits eller at du vil selge PCen din er ikkje god nok grunn til å sende post til studentlistene. Desse er reservert for viktige meldingar. Sender du epost til t.d. sstud med noko uvesentleg, må du huske at 500 personar må lese det du skriv.

### **Påmelding til postlister - majordomo**

Her på instituttet har vi postlistesystemet majordomo. Dette er eit system der ein sjølv kan melde seg på postlister. Dette gjeld til dømes dei fleste kurs over grunnkurs. Dersom du vil melde deg på postlista til i125, sender du epost til majordomo@ii.uib.no , utan tittel, og der einaste innhald i kroppen er ei linje med teksten

```
subscribe i125
```

Da vil du få ei automatisk generert epost melding om at du er med på postlista for kurset, som stadfestar at du er på lista.

## **5.2 WWW**

WWW er kanskje den tenesta på Internet som er best kjend. WWW er såkalla hypertekst. For å lese WWW må du ha ein sidelesar, som kan vise teksten og eventuelle bilete i teksten. Nokre områder i teksten er markert med understreking, og dersom du trykker der, vil du hente ei anna side ein annan stad på nettet. Dei siste åra er sidelesarane utvida, slik at ein kan vise flim (mpeg) , høyre på musikk og radio (realaudio), og køyre program inne i sjølve sidelesaren(java applets).

### **5.2.1 Netscape**

Den vanlegste sidelesaren her på institutt for informatikk, er netscape. For å starte netscape, skriv du på kommandolinja:

```
//bilete av netscape //
```

I det største feltet nede står sjølve teksten. Rett over det er det eit tekstfelt, der du kan skrive inn såkalla URLar. URLar, som og er kjent som Internetadressar, er standard måte å forklare sidelesaren kvar du finn ei side eller annan resurs på Internet. Dersom du skriv ein URL inn i dette feltet, vil netscape vise den sida som URLen peikar til. Dersom du vil se hovudsida for denne manualen, viskar du ut det som står i feltet, og skriv inn

<http://www.ii.uib.no/fu/Handbok/>

Over er det ei linje som gjer at du kan gå tilbake til sist side du var på, og ein knapp for å gå fram igjen. Det er og fleire andre nyttige funksjonar. Det er og mange funksjonar og innstillingar på menyen over.

### 5.2.2 Kva er ein URL

Ein URL er ein standard måte å fortelle ein sideleser kor på nettet ein www-side er. Alle www-sider har ein URL. Til dømes har kurssidene til institutt for informatikk denne URLen:

<http://www.ii.uib.no/kurs/index.html>

URLen satt saman av fleire delar,

### 5.2.3 Sidene til instituttet, kursinformasjon

### 5.2.4 Korleis lage egne heimesider

WWWsider er laga i HTML(HyperText Markup Language). Korleis du kan lage dine egne www-sider kan du sjå på Kjartan sin (litt gamle) introduksjon.

## 5.3 Usenet News

Usenet news(ofte kalla news) er ei stor samling verdsomfattande diskusjonsgrupper om ei rekke ulike emner. Desse diskuterer alt frå operativsystem(t.d comp.os.linux) til kattar ( no.katter ). Gruppene er delt opp i hierarkier, som er ei grovinndeling av gruppene.

no - alle norske grupper comp - Om datamaskinar alt - 'alternative' grupper osv...

Desse kan vere finare oppdelt. Til dømes vil alle comp.lang - gruppene diskutere ulike programmeringspråk.

### 5.3.1 Kva grupper finns

Det er for mange til å gi ein god oversikt her, men dei norske kan du finne på <http://www.usenet.no/>. Ein god oversikt er det og på <ftp://ftp.rtfm.mit.edu/>. På denne anonyme ftp-basen kan du finne dei fleste FAQane (frequently asked questions), for dei gruppene som har det.

### 5.3.2 Ein newslesar: slrn

Ein newsleser som finns på UA er slrn. slrn er tekstbasert, og køyrer i eit terminalvindu. første gang du startar opp slrn, vil

### 5.3.3 Netiquette på Usenet

Dersom du skal poste artikkelar på ein gruppe, er det ein god del reglar å halde seg til. Mange grupper har eit charter, eller ein FAQ (Frequently asked questions) som blir posta med jamne mellomrom. Ellers er det ein generell regel at ein ikkje skal diskutere tema som ikkje har noko med gruppa å gjere, og det vil ikkje aksepterast av dei andre på gruppa.

Tonen på usenet kan vere ganske kvass, men det er ikkje ein god unnskuldning for å vere det sjølv. Skittkasting gjer at informasjonen forsvinn i usakeleg krangel, og det er det ingen som vil.

### 5.3.4 Ftp

FTP er ein måte å hente filer frå eit anna nettverk enn der du er. Det kan enten vere å hente filer frå ei anna maskin enn der du har konto, eller du kan hente offentleg tilgjengelege filer som er lagt ut på såkalla anonym FTP.

### 5.3.5 Hente filer frå ein annan maskin

Dersom du har ein konto på eit anlegg, og vil hente filer til ein annan maskin, kan du bruke ftp til det. Sei du er innlogga som brukar s007 på UA, og vil hente filer som ligg på kontoen stud0001 på it-avdelinga sin maskin alfred.uib.no . Da kan du hente desse filene slik:

```
% ftp alfred.uib.no
Connected to alfred.uib.no
220 alfred.uib.no (Wed Mar 5 12:37:21 EST 1997) ready.
Name (gyro:s007): stud0001
331 Password required for stud0001.
Password: .....
230 User stud0001 logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Du kan hente filer, eller legge inn nye filer på maskina du har ftp-samband med. Dette gjer du med ftp-kommandoar. Dei viktigaste kommandoane er

```
get fil - hent ein fil frå maskina du er inlogga på til lokal maskin
mget fil1 fil2 fil3 ... hent fleire filer samstundes
put fil - flytt ei fil frå lokal maskin til maskina du er inlogga på
mput fil1 fil2 fil3 ... put med fleire filer
cd katalog - gå til ein katalog på maskina du er logga på.
lcd katalog - skift katalog på lokal maskin.
```

Dersom du vil hente filene svada1 og svada2, som er i katalogen foo på maskina du tok ftp til, kan du gjere det slik:

```

ftp> cd foo
ftp> mget svada1 svada2
mget svada1? y
200 PORT command successful.
150 Opening BINARY mode data connection for svada1 (23456 bytes)
226 Transfer complete.
23456 bytes received in 2.3 secs (1.01 Kbytes/sec)
mget svada1? y
200 PORT command successful.
150 Opening BINARY mode data connection for svada2 (1598 bytes).
226 Transfer complete.
1598 bytes received in 0.025 secs (62 Kbytes/sec)
ftp>

```

Det går an å slå av at ftp spør om du verkeleg vil hente ned kvar einaste fil med kommandoen prompt.

### 5.3.6 Anonym FTP, hente offentlig tilgjengelege data

Det finns mykje gratis programvare og anna interessant på nettet. Dette er ofte på anonym ftp. Anonym ftp vil seie at ein kan hente filer frå ein offentlig tilgjengeleg base. i staden for brukarnamnet ditt skriv du 'anonymous' som brukar, og epostadressa di som passord.

Sjå lenger ned om ftpsearch for å finne ut korleis ein kan finne interessante program og anna som er på ftp.

### 5.3.7 Andre FTP-program

Det finns fleire program som prøver å gjere ftp enklare. Dei fleste av desse er først og framst laga for anonym ftp. Blant desse er ftptool, xftp og ncftp. Fleire av desse kan hente ned heile katalogar samstundes. Dette kan ofte vere ein fordel. Som oftast er det ikkje noko problem, sidan det meste på ftpbasar er pakka i tar eller zip-arkiv, som du kan hente som ei enkel fil, og pakke ut til dei filene du vil ha. Sjå kommandolinjekapittelet for ein forklaring av tar.

## 5.4 Søkemotorar

Det er svært mykje informasjon rundt på Internet, både fagleg og svært mykje anna. Sjølv om det finns mykje informasjon, er det ofte vanskeleg å finne det du vil, og kvaliteten varierer. Her vil vi presentere ulike kjelder for å finne informasjon, som kan vere nyttig for informatikkstudentar. Utanom tradisjonelle søkemotorar, vil vi presentere informasjon retta mot studentar

### 5.4.1 Sidene til Institutet

Dersom du vil finne fagleg informasjon, er institutt for informatikk sine heimesider ein bra plass å starte. Her finn du sider for alle kursa på intituttet, og alle

fagområda presenterer seg sjølv. Du finn og lokal informasjon, som tildømes sosiale aktiviteter. Dessutan kan du finne linker til alle studentene sine heimesider. URLen til intituttssidene er:

<http://www.ii.uib.no> - instituttsidene <http://www.uib.no> - iniversitetet sine sider

#### **5.4.2 Boksøk : bibsys**

Dersom du lurer på om universitetsbiblioteka har ei bok om noko du er interessert i kan du bruke bibsys. Det er univeristestbiblioteka sin bokdatabase. Her kan du søke på tittel, forfatter, nøkkelord osv. Universitetsbiblioteket har kurs i bruk av bibsys, sjå oppslag på biblioteket. URLen til bibsys er:

<http://www.bibsys.no/>

#### **5.4.3 Generellt www-søk: Altavista**

Dersom du ikkje har nokon ide om kor du kan finne det du vil ha, kan gjere eit generelt søk på Altavista. Altavista er ein søkemotor som forsøker å ha oversikt over alle www-sider på heile Internet, og har mange millionar sider i registeret.

<http://altavista.digital.com/>

#### **5.4.4 Emnetalogar: <http://www.Yahoo.com>**

Generellt stikkordsøk gir ofte mykje 'boss', sider som ikkje har noko med det du søker etter. Ein litt meir målretta måte å søke på, er å bruke ein emnekatolog. Eit eksempel på ein slik er yahoo. Når du slår opp på hovudsida finn du hovudkategoriar, slik som 'Arts and Humanities', 'Computers and Intrnet' og 'Recreation and Sports'. Her kan du klikke deg vidare til det du er interessert i. Dette er delt enda finere opp i undergruppene, og du kan leite deg fram til tu finn det du er interessert i Dersom du tildømes vil ha informasjon om programmeringspråket java, så velger du

Computers and Internet -> Programming languages -> Java

Dersom du vil vite meir om Edvard Munch kan du velge

Arts and Humanities -> Art History -> Artists -> Edvard Munch

Adressa til Yahoo er <http://www.yahoo.com>

#### **5.4.5 Ftp-søk: [ftpsearch.ntnu.no](http://ftpsearch.ntnu.no)**

Ofte er ein interessert i å finne eit program. Fritt tilgjengelege program er ofte på anonym ftp, og det finns ein eigen søkemotor for ftp på NTNU. Her kan du på same måten som altavista gjere eit generellt søk, men du søker ikkje på www-sider, men prøver å finne filnamn på ftpbasar rund om i verda.

Adressa er <http://ftpsearch.ntnu.no>

#### **5.4.6 Søk etter usenetartiklar: Dejanews**

Dersom du synest å huske at du har lest ein nyttig artikkel på usenet news, men ikkje har teke vare på han, kan du finne han på Dejanews. Dejanews er ein søkemotor spesielt for news. Du kan spesifisere at du vil søke innan ein gruppe

eller eit hierarki. Det går og an å lese og sende newsartiklar på dejanews, men stort sett finns det newsprogramm som er langt betre til dette.

# Kapittel 6

## Meir arbeid på kommandolina

### 6.1 Ymse program

#### 6.1.1 Sova (sleep)

##### Syntaks

```
sleep n
```

#### 6.1.2 Omtale

`sleep` for maskina til å venta i *n* sekund. Du kan til dømes bruka han som ein alarm:

```
t04@bille 1 % sleep 3600 ; echo No må eg hugsa å gå til kantina
No må eg hugsa å gå til kantina
t04@bille 2 %
```

om du veit du har ei avtala om ei time.

Semikolonet (;) skil dei to kommandoane, slik at den andre vert utført når den fyrste er ferdig.

#### 6.1.3 Kalender (cal)

##### Syntaks

```
cal [[månad] år]
```

##### Omtale

`cal` skriv ut ein kalender. Om du ikkje seier anna får du den månadskalenderen som gjeld. Gjev du eit årstal får du kalenderen for heile det året. Ein kan òg beda om kalenderen for ein særskild månad i eit særskild år.

##### Døme

```
t04@bille 1 % cal
August 1998
S M Tu W Th F S
```

```

1
[...]
t04@bille 2 % cal 09 98
    September 98
    S M Tu W Th F S
                    1
[...]
t04@bille 3 % cal 09 1998
    September 1998
    S M Tu W Th F S
                1 2 3 4 5
[...]
t04@bille 4 %

```

## 6.2 Utdata og inndata (< > | (| less))

### 6.2.1 Utdata til fil (>)

#### Ei ny fil

Frå tid til anna, brukar me kommandoar som gjev meir utdata enn det er plass til på skjermen. Då ynskjer med kan henda, å samla dataa i ein fil, eller til eit anna program som kan håndtera dei. Kan henda har du forsøkt å skriva:

```

t04@bille 14 % ls -l /Home/ssstud2
total 464
drwxrwxr-x  5 khalid  i122-adm   512 Apr  8 1997 I122
drwxr-sr-x  2 hans   i160_adm   512 Apr 16 08:26 I160
[...]

```

Der er nok heimekatalogar til å fylgja fleire skjermleNGder. I staden kan me prøva å laga ein fil:

```

t04@bille 18 % ls -l /Home/ssstud2/ > fil-liste
t04@bille 20 % ls -l fil-liste
-rw-r--r--  1 t04      sstud      6863 Aug 11 23:16 fil-liste
t04@bille 21 %

```

Du kan sjå kva som ligg i fila med less(1). Som regel er skalet sett opp slik at du ikkje utan vidare kan overskriva ein gamal fil:

```

t04@bille 7 % ls -l / > fil-liste
fil-liste: File exists.
t04@bille 8 % ls -l / >! fil-liste
t04@bille 9 %

```

I den andre lina her fortel utropsteiknet at me verkeleg ynskjer å skriva over fila. Du kan sjekka korleis fila ser ut mellom kvar kommando.

Av og til får me utdata på skjermen sjølv om me bed om å få dei i ein fil:

```

t04@bille 21 % find /usr/local -name windex -print > filer
find: cannot read dir /usr/local/lib/hershey: Permission denied
find: cannot read dir /usr/local/etc/netman/analyser-1.0: Permission denied
find: cannot read dir /usr/local/etc/netman/etherman-1.1a: Permission denied
t04@bille 22 % cat filer
t04@bille 23 %

```

Unix har to filhendlar for utdata, ein for vanlege data, *stdout* («standard output»), og ein for feilmeldingar, *stderr* («standard error»). Ein enkelt vinkel (>) omdirigerar berre stdout. Om me ynskjer å omdirigera både stdout og stderr, skriv me ">&":

```
t04@bille 26 % find /usr/local -name windex -print > & filer_og_feilmeldingar
t04@bille 27 % cat filer_og_feilmeldingar
find: cannot read dir /usr/local/lib/hershey: Permission denied
find: cannot read dir /usr/local/etc/netman/analyser-1.0: Permission denied
find: cannot read dir /usr/local/etc/netman/etherman-1.1a: Permission denied
t04@bille 28 %
```

Me kan bruka utropsteiknet (!) og ampersangen (&) saman:

```
t04@bille 28 % find /usr/local -name windex -print >&! filer
```

Du kan òg prøva programmet cal(1):

```
t04@bille 37 % cal 98 > kalender
```

### Leggja til sist i ei gamal fil

Me kan leggja nye data til på slutten av ei gamal fil ved å bruka to vinklar:

```
t04@bille 38 % cal 1998 >> kalender
```

(Samanlign dei kalenderane for dei to åra.)

Slik som den enkle vinkelen ikkje vil skriva over ei fil, vil ikkje den doble laga nokon ny om ho ikkje finst:

```
t04@bille 43 % cal >> månadskalender
månadskalender: No such file or directory.
t04@bille 44 % cal >>! månadskalender
t04@bille 45 % cat månadskalender
August 1998
  S  M Tu  W Th  F  S
      1
  2  3  4  5  6  7  8
  9 10 11 12 13 14 15
 16 17 18 19 20 21 22
 23 24 25 26 27 28 29
 30 31
```

## 6.2.2 Utdata til eit program

Om du berre ynskjer å lesa utdataa ein gong, så treng du ikkje laga ein fil og så lesa han med less(1). Du kan senda utdata med less med ein gong. Då brukar me røyret («pipe», |):

```
t04@bille 32 % ls /Home/sstud2 | less
```

Me kan bruka ampersangen saman med røyret òg:

```
t04@bille 34 % man -k vi |& less
```

less(1) er ikkje det einaste programmet, me brukar saman med røyret.

```
t04@bille 34 % cal 1998 | less
t04@bille 35 % cal 1998 | lpr -Plp8
```

## 6.2.3 Inndata frå fil

Det hender òg at me ynskjer å gi eit program inndata frå fil, i staden for å skriva dataa på terminalen. Då brukar me den andre vinkelen (<):

```
t04@bille 36 % mailx -s "testpost" t04 < eksempelfila_mi
```

## 6.3 Oppgåvek kontroll

### 6.3.1 Avbryta eller stoppa ein kommando (Ctrl-Z, Ctrl-C)

Me kan alltid stoppa eller avbryta ein kommando me har starta.

Tast	Engelsk nemning	Norsk nemning	Signalnamn	Signalnr.
Ctrl-C	interrupt	avbryta	SIGINT	2
Ctrl-Z	stop (user)	stoppa	SIGTSTP	24

Når ein kommando vert avbrutt, er han ferdig. Ein kommando som berre vert stoppa kan me starta opp igjen slik at han fortsett der han var. Dette vil verta klarare seinare.

```
t04@bille 4 % sleep 3600 ; echo No må eg hugsa å gå til kantina
^C
t04@bille 5 % sleep 3600 ; echo No må eg hugsa å gå til kantina
^Z
Suspended
t04@bille 6 %
```

### 6.3.2 Bakgrunnsprosessar (&)

I dei eksempla me har sett på til no, køyrer me programma i forgrunnen. Då er det programmet som tar over kontrollen med terminalen, og me får ikkje nytt klarteikn frå skallet før programmet er ferdig.

Av og til ynskjer me å køyre eit program i bakgrunnen, slik at me kan fortsette å bruka skalet til andre ting. Program som treng terminalvindauga for å få inndata, kan sjølvsagt ikkje køyra i bakgrunnen.

Program for X11 kan me gjerne køyra i bakgrunnen, for dei lagar sitt eigne vindaugo og treng ikkje terminalvindauga. Oppgåver som skal køyra i lang tid, utan å trengja noko inndata fra terminalen køyrer me òg ofte i bakgrunnen. Då kan det vera lurt å dirigera utdata til ei fil:

```
t04@bille 8 % xterm &
[2] 23509
t04@bille 9 % cal >& august &
[3] 23517
t04@bille 10 %
[3] Done cal >& august
t04@bille 10 %
[2] Done xterm
t04@bille 10 % sleep 3600 ; echo No må eg hugsa å gå til kantina &
[2] 23519
t04@bille 11 %
```

Når me starter programmet gjev skallet oss to tal. Det fyrste er eit oppgåve-nummeret, medan det andre er prosessnummeret. Me skal bruka både numra seinare. Skalet gjev også melding når programmet avsluttar.

### 6.3.3 Oppgåver (jobs, bg, fg)

Syntaks

```
jobs
bg [%n]
fg [%n]
```

## Omtale

`jobs` gir ein liste over oppgåvene i det skallet du gjev kommandoen. Sett at me nettopp har gjort døma frå dei to siste avsnitta:

```
t04@bille 19 % jobs
[1] + Suspended          sleep 3600
[2] - Running           ( sleep 3600; echo No må eg hugsa å gå til
kantina )
t04@bille 20 %
```

Me kan sjå at den kommandoen me stoppa er her framleis, men han køyrer ikke, han er stoppa («suspended»). Den andre kommandoen køyrer i bakgrunnen («running»). Oppgåvenumra står lengst til venstre. Plussen (+) ved oppgave 1 tydar at det er den aktive oppgåven. Om me hadde hatt eit skal i eit anna vindauga, så ville me ikkje sjå oppgåvene me køyrde der. Kvant skal held berre rede på sine egne oppgaver.

`bg` («background») legger ei oppgave i bakgrunnen, medan `fg` («foreground») legg ho i forgrunnen. Var programmet stoppa, byrjar det å køyra igjen. `bg` og `fg` verkar på det aktive programmet om dei ikke får noko argument. Ellers kan du velja ei særskild oppgave ved å gjeva oppgåvenummeret. Me skriv alltid eit prosentteikn (%) foran eit oppgåvenummer:

### 6.3.4 Prosessar (ps)

#### Syntaks

```
ps [-u brukar]
```

## Omtale

Kvar gong me køyrer et program, startar maskina ein prosess. Medan oppgåvene er knytte til skallet, er prosessane knytte til maskinen. `ps` gjev ein liste over prosessar på maskina. Han kan også gjeva opplysningar om prosessar starta frå eit anna skal.

Kvar oppgave i eit skal er òg ein prosess på maskina. Me hugsar at skalet gav oss eit prosessnummer når me køyrde eit program i bakgrunnen.

Me skil mellom terminalprosessar og ubundne prosessar. Terminalprosessar er knytte til terminalvindaugo som dei er starta frå. Ubundne prosessar er ikkje knytte til nokon terminal.

Me skil også mellom leiarprosessar og endeprosessar. Skalet er sjølvstøtt ein prosess. Når me startar eit program frå kommandolina, vert skalet leiar for den nye prosessen. Endeprosessar er prosessar som ikkje er leiar for nokon annan prosess.

#### Oppsjonar

I utgangspunktet listar `ps` korkje leiarprosessar eller ubundne prosessar. Han gjev heller ikkje så mye informasjon. Du får heller ikkje vita om prosessane åt andre brukarar.

`-a` listar prosessar for alle brukarar, men framleis korkje leiarprosessar eller ubundne prosessar.

- e listar absolutt alle prosessar
- f gjev meir informasjon, til dømes namnet åt brukaren som eig prosessen.
- l gjev meir informasjon, men ikkje den same som «-f»
- u *brukarliste* listar prosessane til dei gjevne brukarane. Brukarane i lista vert skilde med komma, og utan blankteikn.

## Døme

```
Script started on Thu Aug 13 18:56:39 1998
t04@skarabe 1 % ps
  PID TTY      TIME CMD
 24110 pts/4    0:00 tcsh
t04@skarabe 2 % ps -u t04
  PID TTY      TIME CMD
24057 ?          0:00 sleep
24050 ?          0:00 dtterm
23904 pts/2    0:00 tcsh
24018 ?          0:00 xterm
24108 pts/6    0:00 script
[...]
t04@skarabe 3 % ps -l
 F S  UID      PID  PPID  C  PRI  NI           ADDR      SZ      WCHAN  TTY      TIME CMD
 8 R  4984 24110 24109 1  44  20 607b5230    247             pts/4    0:00 tcsh
t04@skarabe 4 % ps -f
  UID  PID  PPID  C    STIME TTY      TIME CMD
   t04 24110 24109 1 18:56:39 pts/4    0:00 sh -i
t04@skarabe 5 % ps -lf
 F S  UID      PID  PPID  C  PRI  NI           ADDR      SZ      WCHAN  STIME TTY      T
IME CMD
 8 R      t04 24110 24109 1  42  20 607b5230    247             18:56:39 pts/4    0
:00 sh -i
t04@skarabe 6 % ps -e
  PID TTY      TIME CMD
   0 ?          0:00 sched
   1 ?          0:17 init
   2 ?          0:02 pageout
   3 ?          66:00 fsflush
[...]
t04@skarabe 7 % ps -ef
  UID  PID  PPID  C    STIME TTY      TIME CMD
  root    0    0  0    May 06 ?          0:00 sched
  root    1    0  0    May 06 ?          0:17 /etc/init -
  root    2    0  0    May 06 ?          0:02 pageout
  root    3    0  0    May 06 ?          66:00 fsflush
[...]
t04@skarabe 9 % ps -af
  UID  PID  PPID  C    STIME TTY      TIME CMD
 georg 24091 24052 0 18:53:36 pts/10  0:00 csh
  t04 23904 23902 0 18:20:59 pts/2    0:00 [ tcsh ]
  t04 24108 24003 0 18:56:39 pts/6    0:00 script ps
  t04 23937    1  0 18:21:01 pts/2    0:00 /usr/dt/bin/ttsession
  t04 23938 23904 0 18:21:01 pts/2    0:00 /usr/dt/bin/dtsession
  t04 24109 24108 0 18:56:39 pts/6    0:00 script ps
  t04 24000 23937 0 18:21:36 pts/2    0:00 /bin/sh -c dtterm -sdtserver
  root 24123 24110 0 18:57:18 pts/4    0:00 ps -af
t04@skarabe 10 %
```

Brukaren *root* er ein systembrukar. Det er instituttet og inga særskild person

som eig kontoen, og alle ingeniørane kan logga inn som *root*. Dei fleste prosessane til *root* vert starta når maskina startar opp, og dei er heilt naudsynt for systemet.

### 6.3.5 Drepa prosessar (kill)

#### Syntaks

```
kill [-signal] oppgåvenummer/prosessnummer
```

#### Omtale

Det hendar at program ikkje verkar som me ynskjer, og me har problem med å få dei til å avslutta. Program som køyrer i forgrunnen kan me som nemd avbryta med Ctrl-C. For å stoppa eller avbryta program i bakgrunnen må me gjera noko anna.

Det som skjer når me trykkjar Ctrl-C eller Ctrl-Z er at skalet sender eit signal til programmet. Kvart signal har eit namn (SIGINT og SIGTSTP) og eit nummer (2, 24). Dei fleste programma tek omsyn til signala og stoppar eller avsluttar når dei blir bedne om det.

*kill* kan òg senda slike signal til prosessane. I utgangspunktet nyttar han SIGTERM («signal to terminate», nr. 15), som verkar omtrent som SIGINT (Ctrl-C).

Som argument til *kill* kan me gjeva både prosessnummer og oppgåvenummer. Oppgåvenummer må innleiast med eit prosentteikn. Prosess nr. -1 vert oppfatta som alle prosessane me eig.

Alle program reagerer ikkje på alle signala. Til dømes vil dei fleste interaktive programma oversjå SIGINT (Ctrl-C), men dei fleste avsluttar når dei får SIGTERM.

Nokre program reagerer ikkje på SIGTERM heller. Det gjeld særleg program som ikke verkar som dei skal og heng seg opp. Det sterkaste signalet me har heiter SIGKILL (nr. 9), og det skal verka få eit kvart program til å avslutta.

Me kan fortelja *kill* kva signal han skal nytta ved å skriva ein minus (-) og namnet eller nummeret til signalet. Namnet skriv me utan forstavinga «SIG». Det einaste signalet me plar bruka på denne måten er SIGKILL. Det er god skikk å alltid prøva SIGTERM før ein prøvar SIGKILL, fordi det gjev programmet høve til å rydda opp etter seg og lagra arbeidet sitt i filsystemet.

#### Døme

```
t04@bille 11 % jobs
[3] + Running                /usr/dt/bin/dtpad
[4] - Running                xt 2
[5]   Running                sleep 600
t04@bille 12 % ps
PID TTY      TIME CMD
27315 pts/7    0:00 sh
27316 pts/7    0:00 dtpad
27307 pts/7    0:00 cxterm
27303 pts/7    0:00 ttsessio
27318 pts/7    0:00 sleep
27305 pts/7    0:00 xt
27242 pts/7    0:00 tcsh
27299 pts/7    0:00 dtpad
```

```

t04@bille 13 %
11e 13 % kill %4
t04@bille 14 % kill -9 %4
t04@bille 15 %
[4] Killed xt 2
t04@bille 15 % kill 27318
t04@bille 16 %
[5] Terminated sleep 600
t04@bille 16 % kill 17316
17316: No such process
t04@bille 17 % kill 27316
t04@bille 18 % Terminated
t04@bille 19 % kill -KILL 27299
[3] Killed /usr/dt/bin/dtpad
t04@bille 21 % kill -KILL 27303
t04@bille 22 % ps
  PID TTY          TIME CMD
 27242 pts/7      0:00 tcsh
t04@bille 23 %

```

## 6.4 Filhandtering

### 6.4.1 Komprimering (gzip, compress, tar, gtar)

#### Syntaks

```

gzip fil [fil...]
gunzip fil [fil...]
compress fil [fil...]
uncompress fil [fil...]
gtar tf tar-fil
gtar tzf gtar-fil
gtar xv|f tar-fil
gtar xv|zf gtar-fil
gtar c|v|f tar-fil fil [fil..]
gtar c|v|zf gtar-fil fil [fil..]

```

#### Omtale

**gzip** og **compress** komprimerar enkeltfiler på disken, slik at dei tek mindre plass. Kvar fil vert byta ut med ein ny fil med endinga «.gz» (gzip) eller «.Z» (compress). **gzip** lagar dei mest kompakte filene. **gunzip** og **uncompress** pakkar filane ut igjen.

Stundom ønskjer me å pakka fleire filer saman i ei. Særleg er det nyttig når me skal komprimera dei. Til dette kan me bruka **gtar**. Me brukar tre hovedefunksjoner, som me markerer med fyrste bokstav i fyrste argument:

```

x extract  pakke ut
c create   samla filer; katalogar vert samla med innhald
t          visa innhald

```

Så kjem eit par oppsjonar:

```

z zip      komprimer den samla fila me gzip(1)
v verbose  skriv filnamna etterkvart som dei vert pakka inn eller ut

```

Til sist kjem ein «f» og så namnet på den samla fila.

`tar` er eit eldre program som verkar som `gtar`, men det manglar «z»-opsjonen.

Det er god skikk å gjeva den samla fila endinga «.tar» om han ikkje er komprimert, eller «.tgz» eller «.tar.gz» om han er komprimert. Stundom finn ein filer som endar på «.tar.Z». Då må ein fyrst bruka `uncompress` og `gtar` etterpå.

## 6.4.2 Program i fleire versjonar (where)

### Syntaks

```
where program
```

### Omtale

Stundom har maskina fleire versjonar av same program. Programmet `where` fortel kvar dei er:

```
t04@bille 3 % where ls
/usr/bin/ls
/bin/ls
/usr/ucb/ls
/local/bin/ls
t04@bille 4 % where netscape
/usr/local/bin/netscape
t04@bille 5 %
```

Me har berre ein versjon av `netscape(1)`, men me kan velja mellom fleire versjonar av `ls(1)`.

```
t04@bille 6 % /bin/ls -l
total 18
drwx-----  2 t04      sstud      512 Jul 31 23:56 Mail
[...]
t04@bille 7 % /usr/ucb/ls -l
total 9
drwx-----  2 t04      512 Jul 31 23:56 Mail
[...]
t04@bille 8 %
```

For å finna ut meir om programma må du lesa man-sidene, og hugsa at nokre program har fleire man-sider. Nokre program har ein opsjon for å fortelja kva versjon det er.

`/bin/ls` og `/usr/bin/ls` er same program, kvifor? Sjekk sjølv med `ls` og hugs kva me lærte om `ln(1)`.



## Kapittel 7

# Programering og kompilatorar

7.1 Kva er eit programmeringspråk og kva er ein kompilator

7.2 Make



## Kapittel 8

# Eige oppsett

### 8.1 Korleis overstyre CDE med eige oppsett

#### 8.1.1 Kva oppstartsfiler finns. Likskap med satsfiler

#### 8.1.2 Kva filer skal ein ta utgangspunkt i



## Kapittel 9

# Reglement for Terminalbruk

### 9.1 Tanken bak reglementet

#### 9.1.1 Netiquette - sunn bruk av internett

#### 9.1.2 Terminalromreglar

### 9.2 Instituttet sitt reglement

### 9.3 Universitetet sitt reglement

