

UNIVERSITETET I OSLO
Institutt for informatikk

UNIX for nybegynnere

Kompendium 43

Dag Langmyhr

Høsten 2005



UNIX for nybegynnere

**Dag Langmyhr
Institutt for informatikk
Universitetet i Oslo**

Høsten 2005

Sammendrag

Dette heftet er ment å være en innføring i bruk av de datamaskinene og det operativsystemet som benyttes i undervisningen her ved *Institutt for informatikk (Ifi)* ved *Universitetet i Oslo*.

Alle datamaskiner har et program som kalles et **operativsystem** til å administrere maskinens ressurser og ta seg av kommunikasjonen med brukerne. Følgelig er dette programmet det første brukerne møter når de skal få kontakt med datamaskinen. På de datamaskinene som er satt av til begynnerundervisning ved instituttet vårt benyttes operativsystemet **UNIX**, så mye av dette heftet vil dreie seg om hva man kan gjøre i UNIX og hvordan man får det til.

Dette heftet ble første gang skrevet i 1988 og er siden revidert flere ganger. Avsnitt som er nye eller endret vesentlig i forhold til den forrige utgaven, er markert med et stort utropstegn i margen (slik som dette avsnittet). **!**

Blindern, 22. august 2005

Dag Langmyhr

Innhold

Innhold	ii
Figurer	iv
Tabeller	v
1 Det første møtet	1
1.1 Skaff deg brukernavn og passord	1
1.2 Finn en maskin	2
1.3 Få kontakt med maskinen	3
1.4 Noen enkle kommandoer	6
1.5 Litt om filer	9
1.6 Hvis du trenger hjelp	10
1.7 Utskrift på papir	12
1.8 Si takk for deg	13
2 Brukergrensesnittet	15
2.1 Ikonraden til venstre	15
2.2 Programoversikten	19
3 Nyttig å vite	21
3.1 Store og små bokstaver	21
3.2 Automatisk fullføring av navn	21
3.3 Noen nyttige programmer for filer	22
3.4 Mer om utskrift	24
3.5 Kompilering	25
3.6 Endring av oppsett	27
3.7 Kart over terminalstuene	28
3.8 Skikk og bruk ved datamaskinen	29
4 Vindussystemet X	33
4.1 Operasjoner på vinduene	33
4.2 Ikoner	36
4.3 VVV og programmet Mozilla	37
5 Filer	39
5.1 Informasjon om filer	39
5.2 Fil-områder	40
5.3 Joker-notasjon for fil-navn	44
5.4 Operativsystemets fil-struktur	44
5.5 Fil-beskyttelse	45
5.6 Linker	49
5.7 Spesielle filer	50
5.8 Automatisk fjerning av filer	52

6 For viderekomne	53
6.1 Tegnsett	53
6.2 Gjenbruk av tidligere kommandoer	53
6.3 Noen nyttige programmer	53
6.4 Omdirigering av I/U	58
6.5 Kommando-filer	62
6.6 Prosesskontroll	63
6.7 Inn- og utlogging	64
7 Når alt går galt	67
7.1 Hjelp fra andre	67
7.2 Problemer med oppsettet	67
7.3 Når filer forsvinner	68
Oversikt over kommandoer	69

Figurer

1.1	Kart over Universitetsområdet på Blindern	3
1.2	Innloggingsvinduet på Linux-maskinene	4
1.3	Standard skjermoppsett ved ifi	5
1.4	Vindu for dagens melding	6
1.5	Et kommandovindu	7
2.1	Forklaring til standard skjermoppsett ved ifi	16
2.2	Brukerens fire delskjermer	16
2.3	Diverse ikoner for CD og diskett	19
2.4	Informasjonen nederst på skjermen	19
2.5	Startmenyen	20
3.1	Automatisk generert kart over terminalstuen i Informatikkbygningen	29
4.1	Vindusmenyen	36
4.2	VVV-leseren Mozilla	37
5.1	Et typisk tre av brukerfiler	41
5.2	Operativsystemets filstruktur	45

Tabeller

4.1	Operasjoner for flytting av tekst	34
5.1	Eksempler på bruk av relativ notasjon for fil-navn	43
5.2	Vanlige umask-koder	49
5.3	Vanlige endelser for filnavn	51
6.1	Tegnsettet ISO 8859-1	54
6.2	Kontrolltaster for redigering av kommandoer	55

For første Gang,
For første Gang,
Det giver mangen Smaating Rang.
— Henrik Wergeland

Kapittel I

Det første møtet

Dette kapittelet er ment å lede en nybegynner gjennom sitt første møte med en av de mange datamaskinene ved instituttet vårt. Følg instruksjonene punkt for punkt, så skulle det bli et problemfritt møte med UNIX og datamaskinenes verden.

1.1 Skaff deg brukernavn og passord

For at datamaskinen i det hele tatt vil ha noe med deg å gjøre, må du skaffe deg et **brukernavn**, et **passord** og en **brukerkonto**.

Brukernavnet er det navnet datamaskinen kjenner deg under, og det vil være bygget opp av bokstaver fra navnet ditt. Alle brukere har sitt unike brukernavn, og dette gjelder på alle datamaskiner ved Ifi og hele Universitetet.

Grunnen til at man har spesielle brukernavn er at det er mye enklere for både datamaskin og bruker hvis alle har korte, entydige brukernavn. Da slipper også maskinen problemer med flere personer med samme navn eller navn som inneholder 'æ', 'ø' eller 'å'.¹

Passordet er din hemmelige adgangskode, og den brukes til å overbevise datamaskinen om at du virkelig er den du gir deg ut for å være. Derfor må du holde dette passordet hemmelig. Hvis en annen får tak i det, vil han eller hun ha adgang til å lese og eventuelt ødelegge alle dine data.

Nye studenter skal få brukernavn og passordet tilsendt i posten når de er tatt opp ved Universitetet. Ifi-studenter kan dessuten få utlevert brukernavn og passord ved ekspedisjonen i 2. etasje i Informatikkbygningen og hos terminalvaktene i de betjente terminalstuene.

For å være helt sikker på at ingen andre vet passordet ditt, bør du endre det så raskt som mulig. Hvorledes dette gjøres er omtalt i avsnittet [Endring av passord](#) på side 27.²

¹ De fleste datamaskiner og operativsystemer er laget i engelsktalende land og der har man ofte problemer med å skjønne at 'æ', 'ø' og 'å' er skikkelige bokstaver.

² I dette skrevet finnes diverse **hyperlinker**. Hvis man leser dokumentet på skjermen, ser man at de skiller seg ut med fargen:

Brukerkontoen er fellesbetegnelsen på de ressursene du har ved Universitetets dataanlegg, som e-postkonto, lagerplass på disk, tilgang til skrivere og lignende. Brukerkonto opprettes automatisk de semestrene du tar kurs.

1.2 Finn en maskin

Utstyrt med brukernavn og passord kan du begi til en av terminalstuene. Ifi har tre slike som er åpne for laveregradsstudenter:

- Den ene terminalstuen befinner seg i kjelleren i Informatikkbygningen («datadrølen») i Gaustadbekkdalen; dette er bygning G04 på kartet i figur 1.1 på neste side. Denne terminalstuen er tilgjengelig i den tiden bygget er åpent (normalt 8.00–20.30; 9.00–13.30 på lørdager).

De som kjenner den hemmelige koden, kan også arbeide her om natten ved å benytte inngangen i det syd-østre hjørnet. Les mer i [Eksempel: Adgangskoden](#) på side 11 om hvorledes du får vite koden.

- Den andre terminalstuen er i kjelleren i Niels Henrik Abels hus («Abels tårn»), bygning 14 i figur 1.1 på neste side. Den er åpen og betjent til samme tider som terminalstuen i Informatikkbygningen.
- I PO-bygningen³ (bygning 16 i figur 1.1 på neste side) bak Niels Henrik Abels hus finnes det også terminalstuer. De er åpne hverdager 10–16.

Dessuten har fakultetet fire terminalstuer hvor også Ifi-studenter har adgang:

- Vilhelm Bjerknes' hus (bygning 13), den gamle lesesalen i 2½. etasje
- Biologibygnngen (Kristine Bonnevis hus, bygning 18), rom 1411
- Fysikkbygningen (bygning 24), fjerde etasje i østfløyen (ved store lesesal)
- Kjemibygnngen (bygning 23), rom VU 49

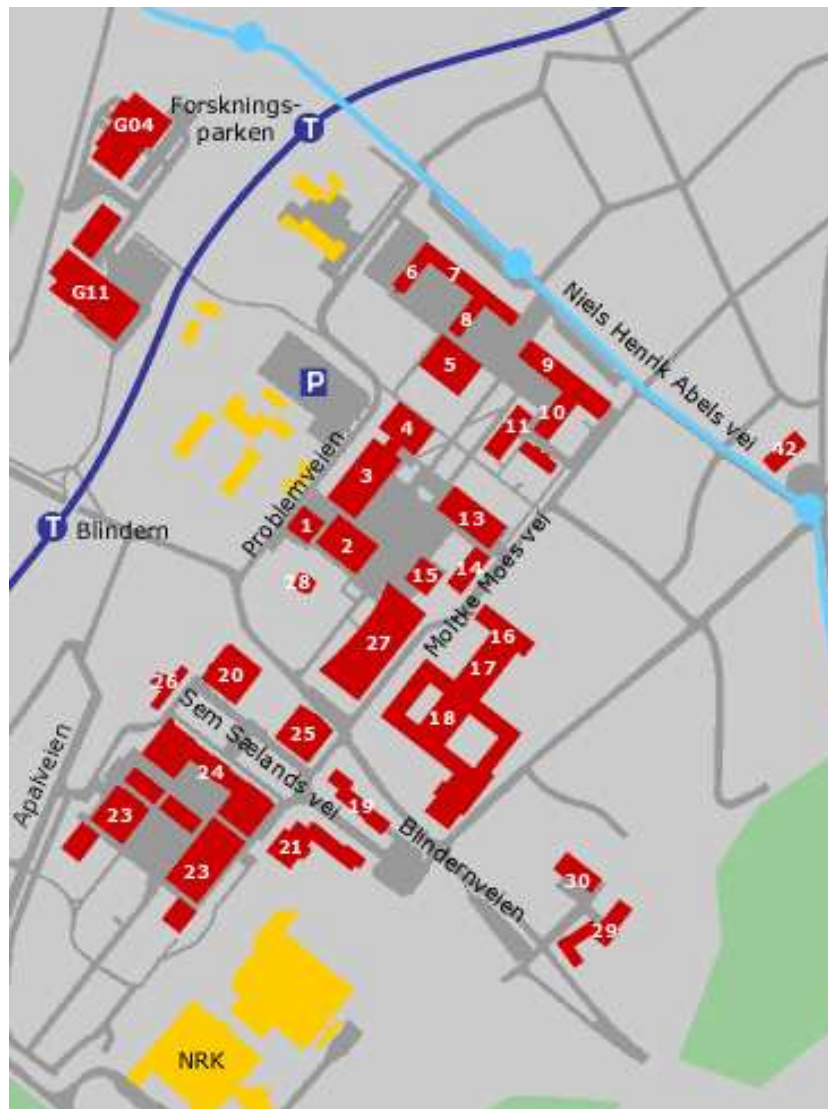
Finn deg en ledig arbeidsplass på en av disse terminalstuene. Det er flere typer datamaskiner å velge mellom så finn en med et skjermbilde som vist i figur 1.2 på side 4.

Hvis det er kø av studenter som vil bruke maskinene, må du skrive deg på en kø-liste og vente til en annen er ferdig.

Interne linker refererer til andre avsnitt og sider i dette skrevet. Ved å peke på en slik link med musen og klikke med venstre mus-tast, flytter man automatisk dit linken angir.

Eksterne linker er til andre dokumenter et eller annet sted ute på Internett. Disse kan man også klikke på, men bare om man sitter ved en datamaskin som er koblet opp mot Internett. Alle datamaskiner ved Ifi er permanent oppkoblet.

³ PO-bygningen har fått dette navnet fordi den tidligere var tilholdssted for *Preklinisk odontologi*.



Figur I.1: Kart over Universitetsområdet på Blindern

På mange av terminalstuene finnes det vennlige **terminalvakter** som kan hjelpe deg med slike praktiske ting. (De kan også mye annet, men det kommer vi tilbake til siden.) Bare spør dem; å hjelpe usikre nybegynnere er en del av jobben deres.

På nettsiden <http://termvakt.uio.no/> finnes mer informasjon om terminalstuene, skrevet av terminalvaktene.

!

I.3 Få kontakt med maskinen

Første skritt er å få kontakt med **innloggingsvinduet**, som ser ut som vist i figur 1.2 på neste side. Hvis skjermen er mørk, flytt litt på **musen** som ligger ved siden av tastaturet.

Hvis du nå ikke har innloggingsvinduet på skjermen, så be om hjelp fra en terminalvakt.



Figur 1.2: Innloggingsvinduet på Linux-maskinene

1.3.1 Gi deg til kjenne

Innloggingsvinduet er en oppfordring til deg om å *logge inn*. (Å opprette kontakt med datamaskinen kalles **å logge inn**; å avslutte kontakten kalles **å logge ut**.) Skriv brukernavnet ditt og trykk på tasten merket eller avhengig av hva slags tastatur du har foran deg.

hunding.ifi.uio.no login: dag

Du skal altså trykke på 4 taster:

hvis du har brukernavn dag.

Skjekk at bokstavene du skriver kommer der de skal, nemlig etter '...login: '. Hvis de ikke gjør det, så trykk en gang eller to på tasten merket eller til skrivemerket (den tynne loddrette streken) står etter '...login: '.

Husk også at du bare skal bruke *små bokstaver* når du skriver brukernavnet ditt, mens passordet kan bestå av både store og små bokstaver. Pass på å skrive begge deler nøyaktig slik du har fått oppgitt. (Det kommer mer om forskjellen mellom store og små bokstaver i avsnittet [Store og små bokstaver](#) på side 21.)

Så langt, så vel. Nå vil maskinen vite passordet ditt. Skriv det, og trykk på eller etterpå. Du behøver ikke være redd for at noen skal titte over skulderen din og se passordet; datamaskinen vil ikke vise de tegnene som hører til passordet.

hunding.ifi.uio.no login: dag
Password: xxxxxxxx



Figur 1.3: Standard skjermoppsett ved ifi

Hvis maskinen nå svarer «Login failed, please try again.» gikk noe galt; enten brukernavnet eller passordet ble skrevet feil. Prøv igjen.

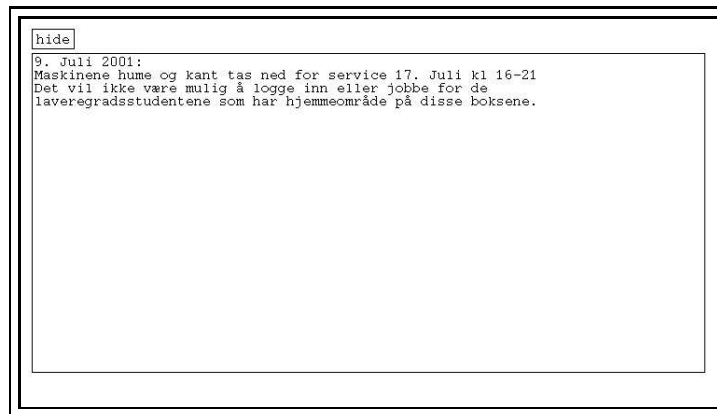
Hvis skjermen blir blank og det etter 5-10 sekunder dukker opp et vindu som ligner på figur 1.3, har du klart det. Maskinen har godtatt deg, og du har kontakt!

Ikke bli fortvilet over alle de rare symbolene du ser på skjermen; de blir forklart i kapittelet [Brukergrensesnittet](#) på side 15.

1.3.2 Dagens melding

Av og til vil du oppleve at det dukker opp et meldingsvindu på skjermen; det ser ut som vist i figur 1.4 på neste side. Dette vinduet inneholder viktig informasjon til alle brukere. Det dukker bare frem når det er nye opplysninger, så les det nøye. Når du har lest det ferdig, kan du fjerne det ved å flytte musen slik at mus-markøren peker på feltet merket **hide**, og så trykke på venstre mus-tast.

Alle brukere forutsettes å lese de meldinger som kommer i meldingsvinduet. De som glemmer dette, vil gå glipp av viktig informasjon.



Figur 1.4: Vindu for dagens melding

1.3.3 Datamaskinenes navn

I innloggingsvinduet står navnet på den maskinen eller X-terminalen du sitter ved ('pusur' i figur 1.2 på side 4). Alle instituttets datamaskiner har navn fordi det er mye hyggeligere enn nummer. Navnene er stort sett hentet fra norsk historie og norrøn mytologi samt italienske byer og landskaper.

1.4 Noen enkle kommandoer

Nå er tiden kommet til at du kan be datamaskinen gjøre noe for deg. Dette skjer ved at du gir maskinen *kommandoer*, og så utfører maskinen disse kommandoene for deg.⁴

Kommandoer gis i et *kommandovindu*. Du får frem et slikt vindu ved å peke med musen på symbolet



og klikke med venstre mus-tast. Da dukker kommandovinduet frem slik det er vist i figur 1.5 på neste side.

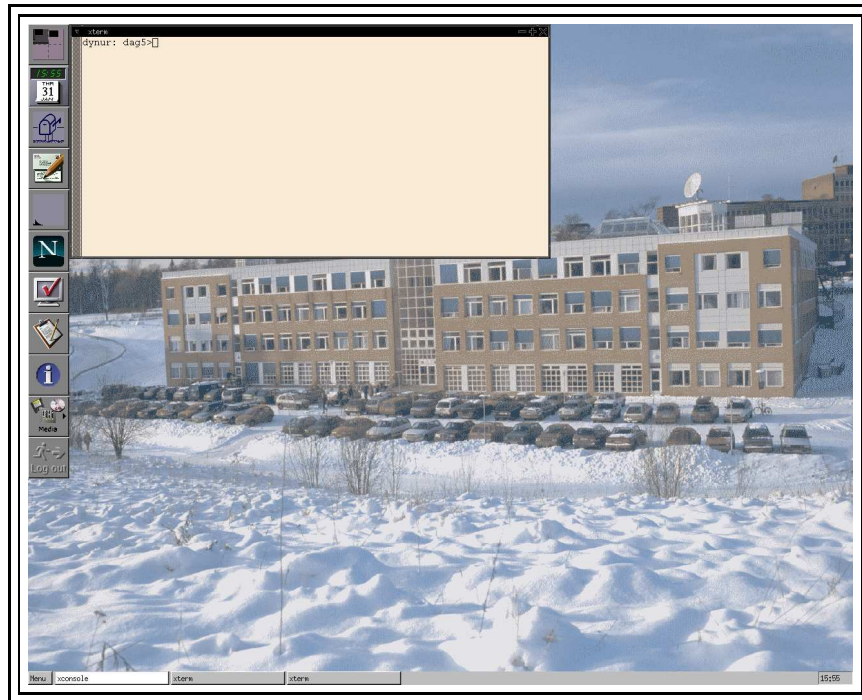
Flytt nå musen slik at du peker på dette kommandovinduet. Hvis du peker et annet sted, vil ikke bokstavene du skriver nå frem dit.⁵

Hver gang maskinen er klar til å motta en ny kommando fra deg, gir den et *klarsignal* (også kalt *ledetekst* eller «prompt» på engelsk):

hunding: dag>

⁴ De to operativsystemene Windows og UNIX har ulik filosofi når det gjelder bruk av tastatur og mus: Windows er mus-orientert mens UNIX er kommando-orientert. Følgelig bruker man tastaturet mye mer når man kjører på en UNIX-maskin.

⁵ Hovedregelen er at tastetrykkene alltid går til det vinduet som musen peker på. Dette er anderledes enn i Windows hvor man vanligvis må klikke på en vindu for å få kontakt med det.



Figur I.5: Et kommandovindu

Klarsignalet angir først navnet på den datamaskinen du kjører på. Etter kolonet står det angitt hvilket filområde brukeren for øyeblikket befinner seg i; det er 'dag' i dette eksemplet. Betydningen av dette feltet omtales nærmere i avsnittet [Manøvrering i fil-områdene](#) på side 42.

I.4.1 Hvilken dag er det i dag?

Aller først kan du spørre datamaskinen hvilken dato det er. Kommandoen for dette heter naturlig nok date:

```
hunding: dag> date  6  
Tue Aug 16 12:28:42 CEST 2005
```

Klokken er altså snart halv ett,⁷ og det er tirsdag den 16. august 2005.

Legg merke til at alle kommandoer avsluttes med .

I.4.2 Retting av feil i kommandoer

Det kan nå være på tide å se litt på hvilke muligheter du har hvis du gjør noe galt, som for eksempel å skrive en kommando feil.

⁶ For å gjøre det helt klart hva brukeren skriver og hva datamaskinen svarer, vil alt som kommer fra datamaskinen skrives med «vanlig skrivemaskinskrift». Det som brukeren taster inn, vil komme med «*skrivemaskinskrift i fet kursiv*» i dette heftet. Dessuten markeres det (med en «  ») når brukeren trykker på tasten merket  eller .

⁷ Tilføyselen CEST står for *Central European Standard Time* og angir hvilken tidssone Norge tilhører.

I.4.2.1 Retting av kommandoer

Hvis du skriver noe galt når du er i ferd med å taste inn kommandoen, kan du fjerne tegn igjen ved å benytte tasten merket  eller ; hvert trykk fjerner ett tegn.

Du kan også fjerne hele linjen og starte på ny frisk hvis du ønsker det; skriv bare  + .⁸

I avsnittet [Gjenbruk av tidligere kommandoer](#) på side 53 står det mer om hvorledes man kan rette feil i kommandoer og gjenta tidligere kommandoer.

I.4.2.2 Å avbryte en kommando

Hvis en kommando av en eller annen grunn løper løpsk, kan du vanligvis stoppe den med å trykke på  + . Av og til er det nødvendig med et par-tre trykk før kommandoen avbrytes. (Det står mer om dette i avsnittet [Å drepe prosesser](#) på side 64.)

I.4.3 En liten kalender

Hvis du vil ha en liten kalender for inneværende måned, finnes det en kommando som lager en for deg:

```
hunding: dag> cal 
      August 2005
Su Mo Tu We Th Fr Sa
  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
```

Du kan også få laget en kalender for en vilkårlig måned, for eksempel for måneden februar i 1994:⁹

```
hunding: dag> cal 2 1994 
Su Mo Tu We Th Fr Sa
      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
```

(Nå er du så vant til at alle kommandoer skal avsluttes med  at det heretter ikke er nødvendig å ta den med i eksemplene.)

Her ga vi ekstra opplysninger til kommandoen om hva vi ønsket, nemlig hvilken måned og hvilket år vi ønsket en kalender for. Slik ekstra informasjon kalles *parametre*, og følger etter navnet på kommandoen, adskilt av blanke.

⁸ Tastekombinasjonen  +  angir et såkalt *kontrolltegn*. Det dannes ved å trykke på tasten merket  eller  og holde den nede mens du trykker på . Et slikt kontrolltegn kan også angis som «Ctrl-C» eller «^C».

⁹ I gamle dager hadde ikke alle land samme kalender; vårt program viser kalenderen som gjaldt i England. Av og til betyr dette litt; bare prøv å se på en kalender for september 1752.

I.4.4 «Ord for dagen»

En annen artig kommando kan gi deg et «ord for dagen»:

```
hunding: dag> fortune
Chisolm's First Corollary to Murphy's Second Law:
    When things just can't possibly get any worse,
    they will.
```

Mange brukere har sørget for at maskinen skriver ut et slikt lite visdomsord hver gang de logger inn eller ut. I avsnittet [Inn- og utlogging](#) på side 64 vil vi komme inn på hvorledes man kan få til noe slikt.

I.5 Litt om filer

La oss nå se litt på begrepet *filer*. En fil er et område på et såkalt *platelager* (oftest kalt *disk*) og det kan inneholde en tekst (som for eksempel et brev) eller et program.

I.5.1 Oversikt over filer

Kommandoen `ls`¹⁰ gir en oversikt over hvilke filer brukeren har på sitt område:

```
hunding: dag> ls
brev   husk
```

I dette tilfellet ligger det altså to filer på området, og de heter henholdsvis `brev` og `husk`. På ditt nye område vil det sannsynligvis ikke være noen filer før du lager dem selv.

Hvis vi ønsker litt mer informasjon om hver fil, kan vi be om det ved å angi en *opsjon*, det vil si en parameter som styrer måten kommandoen utføres på. Det er vanlig i UNIX at opsjoner angis med en '-' foran.¹¹ Eksempelvis gir opsjonen `-l` (for *long*) mer informasjon om den enkelte fil:

```
hunding: dag> ls -l
total 204
-rw----- 1 dag      223 Jan  6 15:23 brev
-rw----- 1 dag      24 Jan  8 12:01 husk
```

Her står navnet på hver fil sist på linjen. De øvrige opplysningene er henholdsvis beskyttelseskode, antall referanser, eier, størrelse samt dato og klokkeslett da filen ble laget. Vi skal komme tilbake med ytterligere tolkning av disse opplysningene siden, nærmere bestemt i avsnittet [Informasjon om filer](#) på side 39.

¹⁰ Navnet `ls` er en litt merkelig forkortelse for *list files*.

¹¹ I Windows bruker man '/' foran opsjonene.

1.5.2 Fremvisning av filer

Vi kan jo se på hva disse filene inneholder; til det benyttes (vanligvis) kommandoen `more` som skriver ut den teksten¹² som ligger i filen, i det vinduet der kommandoen blir gitt.

```
hunding: dag> more husk
Husk forelesning
kl 14.
```

Hvis filen er så stor at det ikke er plass i det aktuelle vinduet, vil `more` vise én skjermfull av gangen.¹³ Trykk ‘`_`’ (blank)¹⁴ for å få neste side eller ‘`q`’ for å avbryte.

1.6 Hvis du trenger hjelp

UNIX er utstyrt med en del hjelpemidler for å gi deg assistanse hvis du står fast. Imidlertid trengs det litt trening i bruk av disse hjelpemidlene før man får fullt utbytte av dem, så sett i gang fortest mulig.

1.6.1 Programmet hjelp

Det viktigste hjelpemidlet for nybegynnere som ønsker hjelp og informasjon om systemet, er programmet `hjel`p som er skrevet ved Ifi. En kort oversikt over noe av det `hjel`p kan gjøre får du ved å skrive

```
hunding: dag> hjelp
Det finnes brukerhjel
```

aplix	drift	ifi	modem	simula
backup	easyio	ijk	oppringt	skrivere
beta	fagutvalget	ijk2	oppsett	staroffice
cdbrenner	framemaker	imap	passord	studentby
cdbrenning	fui	inf-kode	passwd	telefon
debug	grupper	infkode	pop	termvaktfaq
disk-kvoter	hjelp	java	print	universitetet
diskett	hjemmekontor	latex	privatliv	utskrift
diskkvoter	home	maskiner	sigar	windows

Ønsker du `hjel`p om andre emner enn de som er listet opp her, prøv kommandoen «man emne». For å skrive ut en fil, brukes opsjonen `-P«printer»`.

Kommandoen «`hjel`p `hjel`p» gir full beskrivelse av dette.

¹² Kommandoen `more` brukes når vi vet at filen inneholder vanlig tekst. En mer generell kommando er `preview` (se avsnittet [Utskrift av filer på skjermen](#) på side 22); den kan brukes til alle typer filer.

¹³ Det er derfor programmet heter `more`: brukeren får se mer og mer ved å trykke på mellomromstasten.

¹⁴ Tegnet ‘`_`’ benyttes ofte for å betegne et *blankt tegn*, det vil si et tegn som dannes ved å trykke på mellomromstasten, den lange smale tasten nederst på tastaturet.

Vanligvis er det ikke nødvendig å angi slike blanke tegn eksplisitt fordi det går klart frem at de skal være med (for eksempel mellom en kommando og dens parametre), men enkelte ganger – som i dette tilfellet – trenger man å skrive tegnet eksplisitt.

Et kall på hjelp uten parametre (som her) forteller om hvordan kommandoen brukes (det vil si hvilke opsjoner som kan benyttes) og litt om hva den gjør. Den forteller om de (for tiden) rundt 40 emnene¹⁵ som dreier seg om lokale ting ved vårt institutt, og at man ved å skrive

```
hjelp hjelp
```

kan få vite mer om programmet hjelp.

hjelp benytter vanligvis programmet `less`¹⁶ til å gi utskriften. Dette programmet ligner svært på `more`, og brukeren må på samme måte skrive `␣` for å få se neste side og `q` for å avbryte.

I.6.1.1 Eksempel: Adgangskoden

For å kunne komme inn i terminalstuen i Informatikkbygningen, trenger man en adgangskode. Den kan man få vite av hjelp:

```
hunding: dag> hjelp inf-kode
```

Koden til døra utenfor den døgnåpne terminalstua finnes til enhver tid på denne fila. Det vil ikke bli sendt ut spesiell kunngjøring for hver endring. Det betyr at denne fila må leses med jevne mellomrom av alle som ønsker å benytte seg av tilbudet om døgnåpen terminalstue og lesesal. Det er kun studenter ved Ifi som skal ha adgang til dette området. Det betyr at personer som ikke kjenner koden og som ikke kan dokumentere at de er studenter ved Ifi ikke skal slippes gjennom døra selv om de ber om å bli sluppet gjennom.

Den døgnåpne sonen omfatter lesesal og terminalstue samt korridor bortover mot heisen i 1. etasje i Informatikkbygningen. For å ha adgang til resten av bygget må man ha eget nøkkelkort. Ingen som ikke har eget nøkkelkort skal slippes inn i resten av bygningen. Ifis lokaler i PO-bygningen, Abel og Forskningsparken er ikke døgnåpne for andre enn nøkkelkortinnehavere. Laveregradsstudenter som ønsker å bruke terminal etter byggenes stengningstid henvises altså til døgnåpent areal i Informatikkbygningen.

Fra og med 1. mai 2003 etter byggets stengetid er koden `xxxx`.¹⁷

I.6.2 Informasjon om kommandoer

Nesten alle kommandoer har en egen informasjonsfil, og denne kan man få se ved hjelp av kommandoen `man`,¹⁸ for eksempel `man ls` for å få vite mer om kommandoen `ls`.

¹⁵ Noen få emner står oppført under flere navn fordi erfaring har vist at brukerne har problemer med å huske skrivemåten.

¹⁶ Navnet `less` er et eksempel på den litt underfundige humoren man finner en del av i UNIX. Da noen laget et program som lignet på `more`, fant de på å kalle det `less`.

¹⁷ Du må nok bruke `hjelp`-kommandoen for å få vite koden.

¹⁸ Navnet `man` er en forkortelse for *manual*, som blant annet kan bety brukerveiledning.

Disse informasjonsfilene sløser ikke med ordene, og det kreves litt trening for å lese dem. Det beste er å prøve seg på noen av de enkleste kommandoene først, som for eksempel man pwd.¹⁹

1.7 Utskrift på papir

Når du har arbeidet en stund med en tekst eller noe lignende, er det på tide å få en utskrift av det på papir for å ta med hjem og vise dine kjære.

Her ved Ifi skjer utskrift av filer på papir ved å benytte kommandoen print, og den brukes slik:

```
hunding: dag> print fil
```

Det finnes mange skrivere ved instituttet. Som laveregradsstudent har du fem å velge mellom:

inf står i den store terminalstuen i 1. etasje i Informatikkbygningen.

niels og **henrik** står i Niels Henrik Abels hus i et rom ved døren til terminalstuen. Ved å skrive til **abel** vil den komme på enten **niels** eller **henrik**, avhengig av belastningen.

poetica og **ponderosa** står i PO-bygget.²⁰

Kommandoen hjelp skrivere vil alltid gi en oppdatert oversikt over hvilke skrivere vi har og hvem som har adgang til dem.

Normalt kommer utskriften automatisk til den skriveren som er nærmest der du sitter, men du kan velge å sende den til en annen skriver ved å gi opsjonen -P til print:

```
hunding: dag> print -P inf fil
```

Det er terminalvaktens oppgave å hente utskriftene fra skriverne og legge dem i hyllene hvor du kan hente dem.

For å unngå misbruk av skriverne, er det innført en kvoteordning. Dette omtales nærmere i avsnittet [Kvoter på utskrift](#) på side 25.

Det kommer flere opplysninger om print i avsnittet [Mer om utskrift](#) på side 24.

¹⁹ Kommandoen pwd har ikke vært nevnt hittil, men den vil være tema for avsnittet [I hvilket fil-område er jeg?](#) på side 42.

²⁰ Det er ikke tilfeldig at skriverne i PO-bygget har navn som begynner med «po-». Ellers kan nevnes at Poetica og Ponderosa er to kjente skriftsnitt.

I.8 Si takk for deg

Dette kan være nok for første gang. Det er på tide å si takk for seg til datamaskinen. Grunnen til at du så tydelig må fortelle datamaskinen at du er ferdig, er at ikke andre skal kunne sette seg ned ved terminalen og late som de er deg.

Fremgangsmåten for å avslutt sesjonen er følgende:

- 1) Steng alle vinduer du har åpnet; hvis du ikke gjør det, kan du risikerer å miste data.

Kommanduvinduer stenges ved å gi kommandoen 'exit'.

- 2) Klikk med musen på symbolet



og velg «Yes, Really Quit» ved å klikke på det med musen.

Maskinen vil så blanke skjermen og komme frem med innloggingsvinduet igjen. Da vet du at du trygt kan forlate maskinen.

Vel, dette var ditt første møte med en datamaskin. Du er ikke ferdig utlært ennå, men fatt mot: Du er på vei!

Kapittel 2

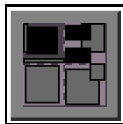
Brukergrensesnittet

Som tidligere nevnt møter brukere ved Ifi et skjermbilde som vist i figur 2.1. I dette kapittelet skal vi se på hvilken funksjon de enkelte feltene og bildene har.

2.1 Ikonraden til venstre

Langs venstre kant av skjermen ser du en samling *ikoner* som gjør det enkelt å starte diverse programmer du vil komme til å bruke ofte. Noen av dem gir også informasjon der de står.

2.1.1 Skjermkartet



Når du bruker oppsettet på Ifis datamaskiner, har du ikke bare én dataskjerm å jobbe på, men fire! Disse står plassert i en 2×2 -formasjon som vist i figur 2.2 på neste side. Skjermkartet viser disse fire delskjermene med alle sine vinduer, men selvfølgelig kraftig forminsket. Det vinduet du ser på for øyeblikket har mørk bakgrunn, de andre har lys.

Den store fordelen med flere delskjermer er det blir enklere å arbeide med flere ting samtidig. Selv bruker jeg for eksempel følgende oppsett:

- 1) Lesing og skriving av e-post.
- 2) Programmering med redigeringsprogrammet Emacs og ett eller flere kommandovinduer.
- 3) Nettlesing med Mozilla.
- 4) Andre programmer.

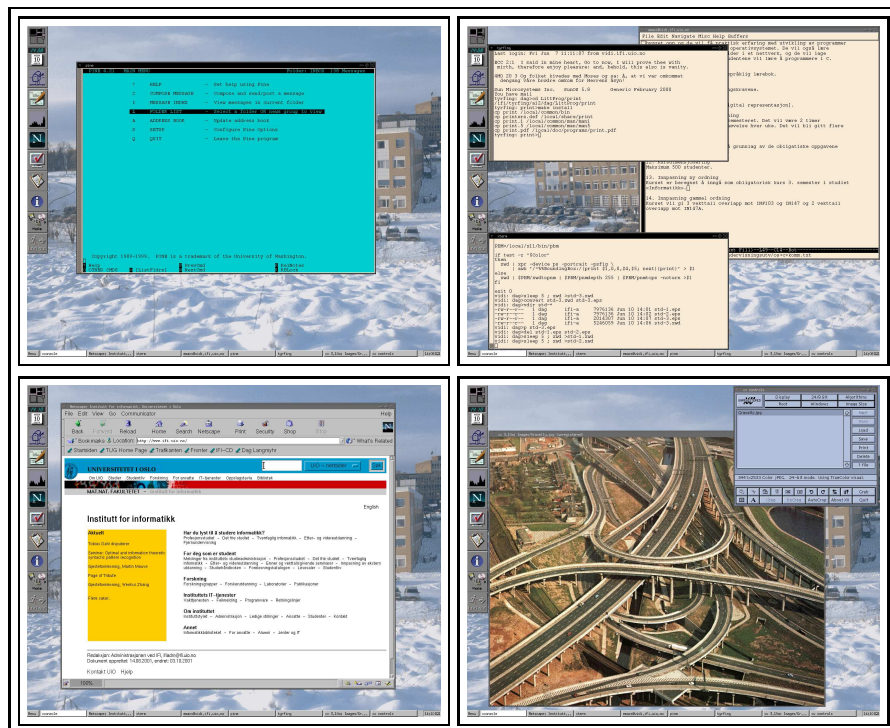
Naturligvis kan man bare se på én delskjerm av gangen. Det finnes to måter å velge hvilken av skjermene du vil se på:

- Du kan bruke musen til å peke og klikke på skjermkartet.
- Du kan velge et program i programoversikten nederst på skjermen (se avsnittet [Programoversikten](#) på side 19); da kommer automatisk riktig delskjerm frem.

KAPITTEL 2 BRUKERGRENSESNITTET



Figur 2.1: Forklaring til standard skjermoppsett ved ifi



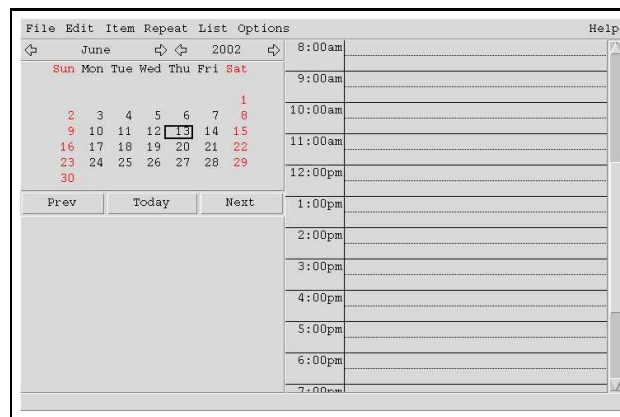
Figur 2.2: Brukerens fire delskjermer

Hint Du kan flytte et vindu til en annen delskjerm ved å dra det ut av én delskjerm og inn på en annen.

2.1.2 Klokken og kalenderen



Dette ikonet er ganske selvforklarende: det viser en klokke og en kalender. (I parentes kan nevnes at klokkene på Ifis datamaskiner går usedvanlig presis; de styres fra et atomur i Tyskland og avviket er normalt betydelig mindre enn ett millisekund.) Kalenderen har dessuten en liten finesse: hvis man klikker på den, kommer det frem en avtalebok:



2.1.3 E-postleseren



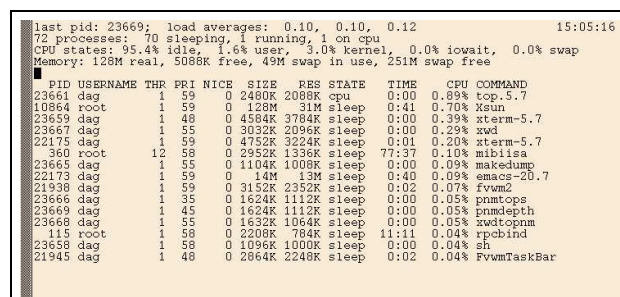
Dette ikonet brukes til å starte et program for å sende og motta e-post. Det finnes mange ulike programmer å velge blant, og brukere har sterke meninger om hvilket som er best. Her ved Ifi er det programmet pine som anbefales og som dermed startes ved hjelp av dette ikonet.

2.1.4 Belastningen



Dette ikonet viser datamaskinens belastning, det vil si hvor mye den har å gjøre. Når kurven er under første strek, bruker ikke datamaskinen all sin kapasitet. Passerer kurven første strek, klarer ikke maskinen alle oppgavene, og programmer går langsommere enn de kunne ha gjort.

Hvis man klikker på dette ikonet, kommer det frem et vindu som gir et mer detaljert bilde av hva maskinen har å jobbe med:



Etter et kurs i operativsystemer vil man forstå hva alle de ulike elementene betyr.

2.1.5 Nettleseren



Dette ikonet starter en nettleser, og på Ifis UNIX-maskiner betyr dette Mozilla. Siden noen nybegynnere prøver å starte Mozilla flere ganger, er det bygget inn en sikkerhetsmekanisme: uansett hvor mange ganger man klikker på dette ikonet, vil det bare bli startet én Mozilla. Ønsker man flere Mozilla-vinduer, kan man få dette ved å velge New ► Navigator Window i File-menyen til Mozilla.

2.1.6 Kommandovinduet



Dette ikonet starter som tidligere nevnt (nærmere bestemt i avsnittet *Noen enkle kommandoer* på side 6) et nytt kommandovindu der brukeren kan skrive sine kommandoer til datamaskinen. Det er ofte man trenger flere kommandovinduer, så derfor er det ikke lagt inn noen sperre på hvor mange kommandovinduer man får lov å ha oppe samtidig; hvert trykk på dette ikonet gir et nytt kommandovindu.

2.1.7 Redigeringsprogrammet



Dette ikonet starter et *redigeringsprogram* hvor du kan skrive og endre programmer, brev, rapporter og annet som skal lagres som vanlig tekst. Ved Ifi benyttes hovedsaklig Emacs til slikt, og det finnes et skriv om *Emacs for nybegynnere* for de som vil lære mer om dette uunværlige verktøyet.

Dette ikonet har også en sikkerhetsmekanisme slik at man ikke kan starte mer enn én utgave av Emacs. Om man ønsker flere vinduer, kan man lett oppnå dette ved å velge find-file-other-frame fra File-menyen i Emacs.

2.1.8 Informasjon

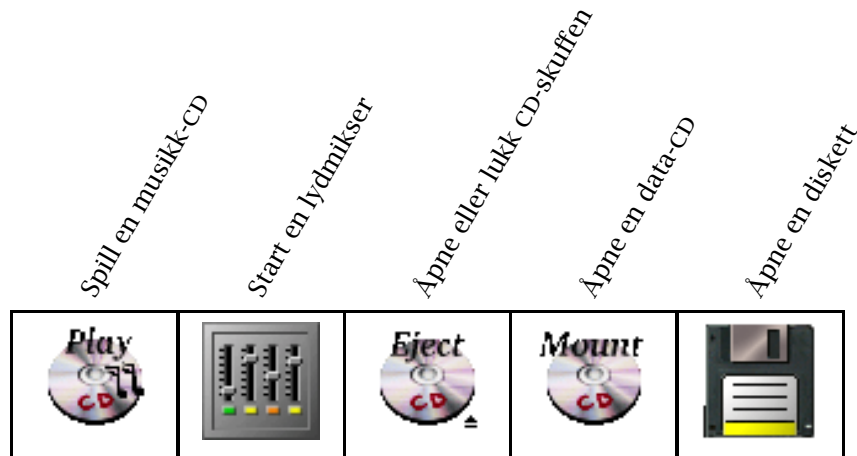


Dette ikonet gir informasjon og hjelp. Når man klikker på det, vil Mozilla bli startet (om det ikke allerede kjører) og det vil automatisk gå til en side som inneholder diverse referanser til hjelp om ulike emner. Forhåpentligvis finner man de opplysningene man trenger der.

2.1.9 Media



Mange av Ifis datamaskiner (men ikke alle) er utstyrt med lydkort, CD-spiller og diskettstasjon; dette gjelder spesielt de maskinene som kjører Linux. Ikonet til venstre gir tilgang til programmer for å styre disse enhetene. Ved å klikke på ikonet kommer det frem fem nye ikoner vist i figur 2.3 på neste side.



Figur 2.3: Diverse ikoner for CD og diskett



Figur 2.4: Informasjonen nederst på skjermen

2.1.10 Utlogging



Når man skal er ferdig med dagens dont og skal si takk for seg til datamaskinen, skjer dette ved å klikke på dette ikonet og så velge «Yes, Really Quit» fra den lille menyen som dukker frem. Fremgangsmåten for å logge ut er ellers beskrevet i avsnittet [Si takk for deg](#) på side 13.

2.2 Programoversikten

Nederst på skjermen finnes det også nyttig informasjon som vist i figur 2.4. Langs det meste av nedre kant står en oversikt av hvilke programmer med egne vinduer som for tiden er i bruk, enten vinduene er åpne eller ikonifisert. (Det kommer et eget avsnitt om [Ikoner](#) på side 36.) Denne oversikten brukes hovedsaklig til to ting:

- Det er lett å se hvilke vinduer som for øyeblikket finnes. Siden vinduer kan befinne seg på ulike delskjerner eller de kan være ikonifisert, kan det være vanskelig å få full oversikt ellers.
- Ved å klikke på et felt i programoversikten vil den tilhørende delskjermen bli valgt og vinduet til det aktuelle programmet vil bli hentet frem om det er dekket av andre vinduer eller det er ikonifisert.

2.2.1 Startmenyen

I nedre venstre hjørne står en meny som gir mulighet til å starte mange flere programmer; når man klikker på den får man frem menyen som er vist i figur 2.5 på neste side. (Man kan også få frem menyen ved å klikke på bakgrunnen, altså utenfor alle vinduene, med venstre mus-tast.) Tegnet «▷» til høyre angir at det finnes en undermeny; om man lar mus-markøren hvile på alternativet, kommer undermenyen frem.

Shells åpner et vindu med en kommandotolker valgt fra menyen med kommandotolkere.

StartMenu	
Shells	▷
Browsers	▷
Text editing	▷
Word processing	▷
Drawing	▷
Spreadsheet	▷
Calculator	
Microsoft Windows	
X Tools	▷
Background	▷
Log Out	▷

Figur 2.5: Startmenyen

Browsers velger blant ulike nettlesere.

Text editing gir et valg blant Emacs og andre redigeringsprogrammer.

Word processing starter programmer for å skrive dokumenter.

Drawing gir tilgang til flere tegneprogrammer.

Spreadsheet benyttes når man trenger et regneark.

Calculator starter en kalkulator.

Microsoft Windows starter et vindu som kjører på en Windows-maskin; på denne måten får man tilgang til programvare som kun fungerer der.

X Tools gir noen spesielle X-programmer.

Background gjør det enkelt å endre bakgrunnen; les mer om dette i avsnittet om [Endring av bakgrunnsbilde](#) på side 28.

Log Out kan brukes nå man vil avslutte kjøringen.

*Det er forbausende hvor få
kunnskaper man klarer seg med i
en moden alder.*

— Nils Kjær

Kapittel 3

Nyttig å vite

Dette kapittelet tar for seg diverse småting som alle UNIX-brukere må kjenne til.

3.1 Store og små bokstaver

I UNIX betraktes store og små bokstaver som forskjellige. Et eksempel på det er:

```
hunding: dag> LS  
LS: Command not found.
```

Feilmeldingen kommer fordi UNIX betrakter `ls` og `LS` som to forskjellige kommandoer hvorav kun den første er definert.

Denne forskjellen på store og små bokstaver er gjennomført i hele UNIX. I alle former for navn, enten det dreier seg om kommandoer, brukere, filer, filområder eller passord, regnes store og små bokstaver som forskjellige.

Det vanlige er at navn skrives med kun små bokstaver, siden dette går raskere. Det eneste unntaket er navn på filer og filområder, hvor man av og til benytter noen store bokstaver.

3.2 Automatisk fullføring av navn

Navn (særlig filnavn) kan ofte bli lange når man benytter UNIX,¹ så det finnes en liten finesse i kommandotolkeren som kan hjelpe brukeren å skrive slike navn.² Foruten at det går raskere, unngår man feiltasting.

Hvis du er i ferd med å skrive et navn på for eksempel en fil, kan du midtveis i navnet trykke på tasten merket  (som på noen tastaturer er merket med ). Da vil systemet fylle ut resten av navnet så langt den

¹ Enkelte operativsystem setter en grense på hvor lange filnavn kan få lov å være, for eksempel 8+3 tegn. I vår versjon av UNIX er grensen 255 tegn.

² Her er det på tide å fortelle at UNIX faktisk har flere forskjellige kommandotolkere, som er programmer som leser kommandoer brukerne gir og sørger for at de blir utført. Hver kommandotolker har sine særegenheter. De tre vanligste heter `sh`, `csh` og `bash` (som er standard ved Ifi).

klarer det, det vil si så lenge resten av navnet er entydig definert. Skriver vi for eksempel

```
hunding: dag> more mi 
```

og minfil.txt er den eneste filen som begynner på mi-, vil systemet fylle ut resten:

```
hunding: dag> more minfil.txt
```

3.2.1 Oversikt over mulige navn

Det finnes en variant av denne finessen som kan liste opp hvilke alternative filnavn som finnes. Her brukes tasten  +  i stedet. Anta at vi skriver

```
hunding: dag> more f  +   
fil1 fil2 fil3  
hunding: dag> more f
```

Systemet gir da en liste over hvilke navn som passer inn i det vi har skrevet, og gjentar linjen slik at vi kan fullføre den.

Advarsel Denne finessen med oversikt over navn bør brukes med forsiktighet, da  +  også vil avslutte en kommandotolker om man gir det som første tegn på en linje! Vær derfor sikker på at du er et stykke ut i en kommando når du skriver  + .

3.3 Noen nyttige programmer for filer

Dette avsnittet omtaler en del nyttige kommandoer for behandling av filer. Et fellestrekk ved dem er at det er kraftige kommandoer med stor mulighet til å gjøre skade. Det er heller ingen angrefrist ved bruk av dem, så man bør vite hva man gjør.³

3.3.1 Oversikt over filer

Kommandoen `ls` benyttes til å få en oversikt over hvilke filer man har. Det finnes eksempler på bruk av kommandoen i forrige kapittel samt en grundig gjennomgåelse i avsnittet [Informasjon om filer](#) på side 39.

3.3.2 Utskrift av filer på skjermen

Som nevnt i forrige kapittel kan kommandoen `more` brukes til utskrift av tekstfiler. Det finnes imidlertid flere ulike formater å lagre filer på, og hvert format krever sitt eget program for å vise innholdet på skjermen. Derfor har vi her ved Ifi laget et eget program for å vise ulike type filer. Programmet heter `preview`, så (omtrent) uansett hva slags fil det er snakk om, vil kommandoen

```
hunding: dag> preview fil
```

³ Dette er også typisk for hele tankegangen bak UNIX; det er et operativsystem for folk som vet hva de gjør.

åpne et vindu som viser innholdet av filen.

Det vil variere litt hva man kan gjøre i et slikt vindu av manøvrering, zooming og annet, men felles for alle de vanligste er at trykk på  (mellomromstasten) eller  benyttes for å bla seg fremover etterhvert som man leser, og  vil avslutte lesingen. For ytterligere informasjon, prøv man preview.

3.3.3 Fjerning av filer

Kommandoen `del`⁴ benyttes til å fjerne filer:

```
hunding: dag> del fil1 fil2
```

Dette er en egen kommando utviklet ved instituttet, slik at brukerne har mulighet til å angre. Den vanlige fjerningskommandoen i UNIX heter `rm`⁵ og den har ingen angremuligheter.

3.3.3.1 Angring etter en fjerning

Hvis man fjerner en fil med kommandoen `del` og etterpå angreir, finnes det muligheter til å hente filen tilbake, i hvert fall inntil man logger ut. Når man først logger ut, forsvinner fjernede filer for godt.⁶

Kommandoen `undel` henter fjernede filer tilbake; hvis man har flyttet seg i filsystemet (se avsnittet [Å flytte til nytt område](#) på side 42) vil filen havne der man for øyeblikket befinner seg. Har man glemt hvilke filer som var fjernet, vil kommandoen `lsdel` gi en oversikt over dem. Den kan gis de samme opsjonene som `ls`.

```
hunding: dag> lsdel
dato fil1 fil2
hunding: dag> undel fil1 fil2
undel: fil1 recovered
undel: fil2 recovered
```

3.3.4 Kopiering av filer

Kopiering av filer skjer ved å benytte kommandoen `copy`.⁷ Man oppgir da først navnet på filen som skal kopieres, og så hva som skal være navnet på kopien.

```
hunding: dag> ls
fil1  fil2
hunding: dag> copy fil2 filny
hunding: dag> ls
fil1  fil2  filny
```

⁴ Navnet `del` er en forkortelse for *delete*.

⁵ Navnet `rm` er en forkortelse for *remove*.

⁶ Heldigvis tas det jevnlig sikkerhetskopier av alle brukeres filer; les mer om dette i avsnittet [Når filer forsvinner](#) på side 68.

⁷ Dette er også en kommando utviklet her ved instituttet for å øke sikkerheten; den protesterer hvis kopiering vil ødelegge en tidligere fil med samme navn. Standardkommandoen for fil-kopiering heter `cp` som er en forkortelse for *copy*.

3.3.5 Endring av navn på filer

Kommandoen `move`⁸ kan benyttes til å endre navnet på en fil. Som parametre oppgir man da først filens nåværende navn og så hva det nye navnet skal være.

```
hunding: dag> ls
fil1  fil2  filny
hunding: dag> move filny fil3
hunding: dag> ls
fil1  fil2  fil3
```

Kommandoen kan også benyttes til å flytte filer mellom filområder. Dette kommer vi tilbake til i avsnittet [Flytting av filer mellom filområder](#) på side 41.

3.4 Mer om utskrift

Som nevnt tidligere får man skrevet ut en fil på laserskriver ved å benytte kommandoen `print`. Det finnes også noen andre nyttige kommandoer som man trenger i forbindelse med utskriften.

3.4.1 Flere opsjoner til print

Foruten opsjonen `-P` til å velge skriver, finnes det mange andre opsjoner som gir brukeren anledning til å kontrollere en utskrift. De viktigste for en nybegynner er følgende:

- P skriver** velger hvilken skriver man vil benytte (hvis det skal være en annen enn standardskriveren).⁹
- p nn:mm** velger hvilke sider som skal skrives ut, hvis man ikke ønsker hele filen.
- multi2** forminsker hver side til halve størrelsen og man får dermed plass til fire sider på hvert ark (to på forsiden og to på baksiden). Det finnes også en `-multi4`, en `-multi8`, osv.
- at tt:mm** angir tidspunktet filen skal skrives ut. Hvis det ikke haster med utskriften, er det hensynsfullt overfor de andre brukerne å vente med utskriften til natten.

Hvis man for eksempel gir kommandoen

```
hunding: dag> print -P inf -p 1:4 -multi2 -at 2:00 husk
```

betyr det at filen `husk` skal skrives ut på `inf1` eller `inf2` klokken 2 til natten. Jeg er bare interessert i de fire første sidene, og jeg vil ha dem i halv størrelse.

En full liste over alle opsjonene får man ved å skrive `man print`.

⁸ Dette er enda en kommando laget her ved Ifi, og grunnen er også her sikkerheten. Standardkommandoen for omnavning av filer heter `mv` som er en forkortelse for `move`.

⁹ Her ved instituttet har vi et opplegg hvor den skriveren som er nærmest der man sitter, automatisk velges som standardskriver. Dette skjer når du logger inn.

Hvis du lurer på hvilken skriver som er valgt, kan du gi kommandoen `printenv PRINTER`.

3.4.2 Kvoter på utskrift

Universitetet har bestemt at alle studenter skal betale for det de skriver ut på universitetets laserskrivere. Dette er utførlig beskrevet på nettsiden [//www.usit.uio.no/it/lita/utskriftsbetaling.html](http://www.usit.uio.no/it/lita/utskriftsbetaling.html). !

På <https://www.uio.no/usit/brukeradm/> kan du finne ut hvordan du ligger an og hvor mange ark du har igjen.

3.4.3 Printerkøen

Når man har foretatt en utskrift med kommandoen `print`, blir den lagt i en kø til skriveren er ledig. Ved å benytte kommandoen `lpq`¹⁰ kan man se hvor mange som ligger i køen.

```
hunding: dag> lpq
inf:
Rank  Owner   Job Name                               Size    Time
active per    958 /tmp/guide.dvi@vit      229699 17:18:46
2     dag     987 /tmp/Prog.java@logi      69974 17:18:47
3     anneber 12  /tmp/Info.txt@tyrfing     12699 17:18:50
```

Min utskrift er altså nummer 2 i køen.

3.4.4 Fjerning fra utskriftskøen

Hvis jeg plutselig finner ut at jeg ikke ønsker utskriften allikevel, kan jeg fjerne den fra køen ved å benytte kommandoen `lprm`:¹¹

```
hunding: dag> lprm 987
inf:
dequeued job 987 from hunding.ifi.uio.no
```

Når man fjerner en utskrift, må man oppgi dens nummer. Det finner man under ordet `Job` når man bruker kommandoen `lpq`.

3.5 Kompilering

Mye av opplæringen ved instituttet dreier seg om å skrive programmer, kompilere¹² dem og utføre dem. Det finnes kompilatorer for de fleste vanlige språkene: JAVA, PERL, PYTHON, RUBY, LISP, C og C++.

3.5.1 Java-kompilatoren

Java-programmer kompileres med kommandoen `javac`.

```
hunding: dag> javac Primes.java
```

Hvis alt går bra og kompilatoren ikke finner noen feil, kommer det ingen utskrift, bare et nytt klarsignal.

¹⁰ Navnet `lpq` er en forkortelse for *line printer queue*.

¹¹ Navnet `lprm` er en litt fordreid forkortelse for *line printer queue remove*.

¹² Å *kompilere* et program betyr å oversette det til en kode som kan utføres på datamaskinen. Programmet som foretar oversettelsen kalles en *kompilator*.

3.5.2 C-kompilatoren

Når man skal kompilere et C-program benyttes kommandoen `cc` eller `gcc`. Hvis det ikke gis noen spesielle opsjoner, vil den kompilere et C-program og lage en kjørbart fil med navn `a.out`; derfor brukes omtrent alltid opsjonen `-o` for å angi navnet på filen der det ferdige programmet skal legges.

```
hunding: dag> cc primes.c -o primes
```

Ytterligere informasjon om kommandoen og hvilke opsjoner som kan benyttes, får man ved å skrive man `cc`.

3.5.3 Kompilering fra Emacs

Redigeringsprogrammet Emacs inneholder en mulighet for å kompilere programmer uten å gå om kommandovinduet. Dette omtales nærmere i *Emacs for nybegynnere*.

3.5.4 Hvordan kjøre et ferdig compilert program

Det er enkelt å kjøre programmet når det er ferdig compilert; det er bare å benytte navnet på filen med det ferdigkompilete programmet som en kommando.¹³ Følgelig startes programmet ved å skrive

```
hunding: dag> primes
      2   3   5   7  11  13  17  19
```

3.5.4.1 Forskjellige maskintyper

Når man skal kompilere og kjøre egne programmer, må man være bevisst på at det finnes flere maskintyper her ved instituttet; de aller fleste er av merket SUN (som kjører en UNIX-variant som heter *Solaris*) eller ulike varianter av PC-er (som kjører UNIX-systemet *Linux*).

Problemet er at et program compilert på én maskintype ikke kan kjøres på en annen maskintype. Sitter man altså en dag og jobber ved en SUN-maskin og neste dag setter seg ved en PC, må man kompilere på nytt alle de programmene man skal kjøre.

3.5.4.2 Hvordan kjøre et compilert Java-program

Java-programmer er litt spesielle; der må man bruke kommandoen `java` for å få kjørt den kompilerte koden:

```
hunding: dag> java Primes
      2   3   5   7  11  13  17  19
```

En annen spesiell ting ved Java-programmene er at den kompilerte koden er den samme for alle maskintyper. Man trenger følgelig ikke rekompilere koden om man skifter maskin.

¹³ Dette gjelder faktisk omtrent alle kommandoene i UNIX. De er programmer (vanligvis skrevet i programmeringsspråket C) som kjøres når vi nevner navnet på dem.

3.6 Endring av oppsett

UNIX-oppsettet ved Ifi er laget slik at det gir en god arbeidsomgivelse for nybegynnere, mens mer erfarne brukere lett kan endre det slik de ønsker det.

3.6.1 Endring av passord

Hvis du lurer på om noen har fått greie på passordet ditt, er det på tide å bytte det. Husk at en person med kjennskap til passordet ditt kan gjøre stor skade, for eksempel fjerne alle dine filer eller sende slibrige brev i ditt navn. For å være på den sikre siden kan det være lurt å skifte passordet med en gang du får en konto, og derefter 2–4 ganger i året.

Regn ikke med at det passordet du har fått utdelt, er hemmelig; skift dette passordet så fort som mulig!

Husk at passordet er personlig og ikke skal «lånes ut» til andre; kontoen vil bli sperret hvis dette skjer.

Når man skifter passord, er det fristende å skifte til et som er lett å huske. Ulempen er at passord som er lette å huske, ofte er lette å gjette for andre. Unngå særlig bruk av eget eller kjærestens navn, fødselsdatoen eller lignende. Passordet må være på 8 tegn; bruk gjerne noen store og noen små bokstaver samt et siffer og et «rart» tegn (som «\$» eller «/»), så blir det mye vanskeligere å gjette.

Tips En lur måte å velge passord på er å tenke på et ord man forbinder noe med, og så skrive dette med innebygde «feil» og «bokstavbilder». Anta at man har vært på ferie i **Neandertal** (en vakker liten dal øst for Düsseldorf); da kan man velge passordet «**ne&=1234**» med følgende forklaring:

ne	
&	and (den engelske betydningen)
=	er (i regnestykker som $1 + 1 = 2$)
1234	noen tal(l)

Om man bruker litt fantasi, er slike passord rimelig lette å huske og omtrent umulig å gjette.

Selve skiften av passordet skjer ved å benytte Internett-siden <https://www.uio.no/cgi-bin/usit/ureg2000/password>. Her kan man fylle ut brukernavn og passord (både det gamle og det nye) og så sende dette inn. !

Systemet vil sjekke det nye passordet for å se om det er lett å gjette. Dette skjer blant annet ved å sjekke diverse ordlister, både norske og engelske. Hvis passordet er for enkelt, vil systemet nekte å godta det, så du må finne et annet.

Etter at du har endret passordet, vil det ta noen timer før hele systemet ved Universitetet er klar over endringen. Derfor lønner det seg å bytte passord rett før du skal avslutte dagens arbeid og gå hjem.

Hvis du glemmer passordet ditt, er du ille ute, for systemet er slik at ingen skal kunne finne frem til det. Ta derfor godt vare på det.¹⁴

3.6.2 Endring av bakgrunnsbilde

Selv om Ifi-bygningen er pen å se på, kan det tenkes at noen ønsker en annen bakgrunn. Det er heldigvis svært enkelt å endre bakgrunnsbilde: man starter bare med menyen i nedre venstre hjørne av skjermen og velger Backgrounds ► Pictures ►¹⁵ *det bildet man liker*. Man kan alternativt velge et bakgrunnsmønster eller en ensfarget bakgrunn.

3.6.2.1 Bruk av eget bilde

Hvis man har et eget bilde¹⁶ man vil ha som bakgrunn, må dette legges et spesielt sted:

```
hunding: dag> mkdir ~/.ifi17
hunding: dag> mkdir ~/.ifi/background
hunding: dag> copy mittbilde ~/.ifi/background
```

! Hvis det ligger flere bilder i denne mappen, vil systemet velge ett tilfeldig hver gang du logger inn.

3.7 Kart over terminalstuene

! Her ved Ifi er det utviklet programmer som automatisk tegner opp kart over terminalstuene, slik som vist i figur 3.1 på neste side som er laget av infmap. For hver datamaskin står det angitt hva den heter og **** om den er opptatt.

Kommandoene for å lage dette kartet heter 121map, 1302map, 1303map, 2406map, 3217map, abelmap, abelvaktmap, antarktismap, baktusmap, bibmap, biomap, brakkemap, buretmap, c207map, c210map, cybmap, disectmap, driftmap, dsbmap, eplemap, fargemap, fp2map, fui-verdandemap, fuimap, fysmap, himalayamap, indiamap, indyemap, infmap, infwidemap, jensmap, kariusmap, kjemimap, meddm2map, meskursmap, mmclmap, parkmap, pingmap, poboxmap, pomap, printerrommap, rsumap, scimap, snittmap, solmap, tannhullmap, terramap, testmap, texasmap, vbmap, veilabmap, verdandemap, vindumap, vlsimap og yamamap.

¹⁴ Skulle du miste det, er ikke alt tapt allikevel. Driftspersonalet kan skaffe deg et nytt; ta kontakt med dem.

¹⁵ Det tar litt tid å lage de små eksempelbildene, så etter å ha pekt på Pictures tar det 5–10 sekunder før alternativene dukker frem.

¹⁶ Det er ikke alle bilder som egner seg som bakgrunnsbilde. For eksempel blir bildet gjengitt med maksimalt 96 ulike farger på noen av våre skjermer, og enkelte bilder blir ødelagt av dette. Men bare prøv – det er trivielt å velge et annet bilde etterpå.

¹⁷ Denne mkdir-kommandoen skal selvfølgelig ikke gjøres om området finnes fra før, men det gjør det neppe for nye brukere.

+ - w - - - - - + - w - - - - - + - - E U R O P A - - + - - - - - + - - - - - +													
portopalo		predazzo		dvalinn		trepani		portogrua		ribera			
								GONE					
- - - - - + - - - - - ^ - - - - - + - - - - - ^ - - - - - + - - - - -													
- w - - - - - + - - - - - w - - - - - + - - - - - + - - - - - + - - - - - +													
+ - - - - - +		alia		goito		marcara		****		stilo		vestri	
melza		+ - w - -		CORSICA - - w - +		+ - - - -		SICILIA - - - - +		+ - - - -		KRETA - - - - +	
		nyi		castrofil		ai		mede		longarone		torretta	

- - - - - + - - - - - - - - - - + - - - - -													
+ - +										+ - - - -		+ - - - - +	
TERM-		- - - - -		- - - - -		- - - - -		+ - - - -		****			
VAKT								DOWN		usini		lampedusa	
+ - - - - +		marineo		falcone		pizzeria		burgio		- - - -		KYPROS - - - -	
- - - - +		+ - - - -		SARDINIA - - - - +		+ - - - -		MALTA - - - - +		orzINUOVI		motsognir	
BRUS		bosa		northri		altjofr		cammarata					
- - - - +				****						+ - - - -		+ - - - - +	
										austri		panteller	
- - - - - + - - - - - - - - - - + - - - - -													
+ - - - - - + - - - - - + - - - - - v - - - - - + - - - - - v - - - - - + - - - - - + - - - - - +													
				****						****			
assoro		suthri		durinn		brucoli		palmanova		lendinaro		nithi	
+ - - - - +		+ - - - - +		+ - - - - +		+ - - - - +		A E R T K A - - - - +		+ - - - - +		+ - - - - +	

3.8.1 Ikke snok i andres filer

Folk skriver så mye rart i sine filer, og de færreste ønsker at alle skal kunne lese sine private ting. Derfor finnes det beskyttelsesmekanismer i UNIX, men det hender at folk ikke bruker dem skikkelig. I et system av vår størrelse vil det alltid finnes filer som burde vært beskyttet bedre enn de er.

Regelen er at man ikke skal lese filer på en annens område med mindre det dreier seg om et nærmere avtalt samarbeid. Det å lete i andres områder etter dårlig beskyttede filer er helt forkastelig, enten det dreier seg om andre brukeres områder eller systemområder.

Skulle du finne en forlatt maskin hvor brukeren ikke er logget ut, så ikke begynn å titte på vedkommendes filer. Situasjonen har mye til felles med det å komme inn på et ulåst kontor; selv om papirer ligger åpent på skrivebordet, er det ikke god tone å lese dem. Det korrekte (når det gjelder datamaskiner) er å logge brukeren ut.

3.8.2 Ikke sløs med diskplassen



Alle vil før eller siden i løpet av semesteret oppdage at maskinen ikke har ubegrenset med diskplass, det vil si lagerplass for filer. Laveregradsstudent får derfor unngå å bruke mer enn 25 Mbyte diskplass.

For å sjekke hvor mye diskplass du bruker, kan du bruke kommandoen `du`¹⁹:

```
hunding: dag> du -s ~20  
1264
```

I dette eksemplet bruker jeg 1264 Kbyte eller litt under 1,3 Mbyte. Det er altså et godt stykke igjen til jeg når grensen.

Den enkelte bruker må derfor hele tiden prøve å begrense sitt plassforbruk. Dette innebærer blant annet at man jevnlig fjerner unødvendige filer; har man for eksempel kildefilen, trenger man ikke oppbevare det ferdigkompilete programmet. En annen mulighet er å komprimere filer; det omtales i avsnittet [Komprimering av filer](#) på side 57.

3.8.3 Vis ansvar når du sender e-post

Studenter ved Universitetet i Oslo kan sende elektronisk post til hvem de vil i hele verden. Dette systemet er basert på tillit til at det ikke misbrukes. Unngå derfor å belaste postsystemet med dillete eller fornærmelige innlegg. Dette gjelder spesielt post som går til mange mottakere, som nyhetsgrupper og «pratefora» (som IRC).

¹⁹ Navnet du står for *disk usage*.

²⁰ Denne kommandoen forteller ikke hele sannheten; den oppgir bare hvor mye som er lagret på mitt private område. Det kan tenkes at jeg i tillegg har lagret filer på andres områder eller at det er mye ulest e-post; slike filer teller også med i brukerens kvote.

3.8.4 Vis ansvar på Internett-siden din

Alle brukere ved Ifi har anledning til å lage seg sin egen hjemmeside på Internett. Denne blir da tilgjengelig for vvv-lesere i hele verden. Vis derfor hensyn, og legg ikke bilder og annet materiale der som folk kan finne støtende.

3.8.5 Ikke prøv å ødelegge systemet

Det er mulig å gjøre et operativsystem enten brukervennlig eller sikkert, men ikke begge deler. Her ved instituttet har vi stort sett satset på å gjøre systemet åpent og brukervennlig. Det innebærer at det er fullt mulig for vanlige brukere å trenge igjennom sikkerhetsmekanismene og skaffe seg kontroll over systemet. Det er en sjanse vi må ta hvis vi ønsker gode arbeidsforhold for alle, og driftsgruppen har tillit til at dette ikke misbrukes.

Hvis noen blir grepet i å prøve å ødelegge systemet eller tilta seg privilegier de ikke har rett til, vil vedkommende bli utestengt fra anlegget for kortere eller lengre tid.

3.8.6 Ikke ødelegg for andre

En annen følge av at vi har satset på et åpent system, er at det er mulig å ha det moro med andre brukere, for eksempel ved å sørge for at det skjer overraskende ting på skjermen deres. Slik bruk regnes som misbruk, og i alvorlige tilfeller kan det føre til utestengning fra anlegget.

3.8.7 Skill mellom lek og arbeid

På maskinene ved Ifi finnes det endel spill, og det er ingenting i veien for å prøve disse en sen kveldstid. Imidlertid må man være oppmerksom på at spill har absolutt lavest prioritet av all virksomhet. Bruk derfor bare spillene hvis det er en tid på døgnet hvor det finnes mange ledige maskiner.

På Internett finnes det nyttig stoff, faglig stoff og masse annet, og mange har som hobby å «surfe» på Nettet. Den slags aktivitet er på samme måte helt i orden, så lenge den utføres på tider da det er få andre på terminalstuene.

Hvis du skal arbeide seriøst og ikke finner noen ledig maskin, men oppdager en slusk som sitter med ett eller annet spill eller leser ikke-faglig stoff på Internett, har du full rett til å kaste vekk ham eller henne øyeblikkelig. Du kan også ta kontakt med terminalvaktene; de kaster mer enn gjerne vekk slike folk.

Er da det du ser gjennom dette
vinduet mot verden så vakkert at
du ikke ønsker å skue gjennom
noe annet vindu?
— Friedrich Nietzsche

Kapittel 4

Vindussystemet X

Det er nå på tide å se litt nærmere på de mulighetene som vindussystemet X på UNIX-maskinene våre tilbyr.

Alle våre dataarbeidsplasser er utstyrt med grafisk skjerm og mus. På disse skjermene kan det vises flere vinduer samtidig. For å kunne gjøre dette, trengs et **vindussystem**, et program som kan opprette vinduer og holde orden på dem. Her ved instituttet benytter vi vindussystemet **X** i versjon 11.6; det omtales vanligvis som **X11**.

Vi har dessuten et annet program som brukeren benytter til å flytte vinduer på skjermen, endre størrelsen på dem, ikonifisere dem og lignende. Det kalles en **vindussjef** (på engelsk «window manager»). Standard vindussjef for våre UNIX-maskiner heter **fvwm**.¹

4.1 Operasjoner på vinduene

4.1.1 Musen

Maskinene våre er utstyrt med en **mus** til å peke med. Hvis du flytter den, vil du se at et merke på skjermen flytter seg. Dette merket kalles **mus-markøren**, og det skifter utseende etter hvilket vindu det befinner seg i; det vanligste er at det ser ut som en skråpil (↖). Hvis mus-markøren ikke er i noe vindu ser den ut som et kryss (×).

Musen er også utstyrt med 3 taster til å trykke på. Hvilken effekt dette har vil variere med hvilket vindu musen² befinner seg i.

4.1.2 Fokus

På skjermen finnes det altså mange vinduer som fungerer som små skjermer, men maskinene har bare ett tastatur. Dette problemet er løst ved at man bruker musen til å velge hvilket vindu som skal være i **fokus**. Når man trykker på en tast, vil det tilhørende tegnet bli sendt til det

¹ Ingen vet hva bokstavene *fvwm* står for. Opphavsmannen selv sier: «The name *FVWM* used to stand for something, but I forgot what.»

² Ofte sier vi *musen* når vi egentlig mener *mus-markøren*. Dette gir imidlertid sjelden opphav til misforståelse.

Tastetrykk	Effekt
<i>⟨venstre mus-tast⟩</i>	Velger ut tekst som skal legges i utklippsbufferen. Pek først på den ene enden av teksten; trykk så på tasten og flytt mus-markøren til den andre enden av teksten. (Den utvalgte teksten blir da invertert.) Slipp så tasten. Teksten blir ikke fjernet fra vinduet, bare kopiert over i utklippsbufferen.
<i>⟨midtre mus-tast⟩</i>	Setter teksten fra utklippsbufferen inn i vinduet. Hvor mus-markøren peker har nå ingen betydning annet enn å velge fokus: i et kommandovindu kommer teksten der neste kommando skal skrives, i et Emacs-vindu kommer den der Emacs-markøren peker.
<i>⟨høyre mus-tast⟩</i>	Utvider innholdet av utklippsbufferen med det som ligger mellom forrige valg og der mus-markøren nå befinner seg.

Tabell 4.1: Operasjoner for flytting av tekst

vinduet som er i fokus, altså det som musen peker på. Hvis musen peker på bakgrunnen når man skriver et tegn, vil tegnet bare forsvinne.

Det vinduet som er i fokus, kalles også for det **aktive vinduet** og dette er markert ved at vinduet har en annen farge på rammen enn de andre. Alle tastetrykk som brukeren gjør, går til det programmet som «eier» dette vinduet.

Et vindu kan godt være i fokus selv om det er delvis dekket av ett eller flere andre vinduer; det vesentlige er at mus-markøren peker på vinduet. (I avsnittet [Vinduer over og under hverandre](#) på denne siden står det fortalt hvor man løfter frem et vindu som er dekket av andre.)

4.1.3 Kopiering av tekst

X11 har mulighet til å kopiere tekst innen et vindu eller mellom vinduer. Man kan for eksempel kopiere linjer fra Emacs-vinduet til et kommandovindu eller fra ett kommandovindu til et annet.

Kopiering av tekst skjer via en egen utklippsbuffer i X11. Operasjonene på denne bufferen er gitt i tabell [4.1](#). (Legg merke til at du ikke trenger å utføre noen kommandoer eller menyvalg som du må i Windows.)

4.1.4 Vinduer over og under hverandre

Hvis man oppretter flere vinduer på skjermen, vil man oppleve at vinduer dekker hverandre. Analogien er et skrivebord, hvor det ofte ligger mange ark. Noen ark vil da ligge oppå andre, og disse vil helt eller delvis dekke de som ligger under.

Ønsker vi å se nærmere på et vindu som ligger under et annet, kan vi med musen peke på toppfeltet eller rammen i det underste vinduet og klikke med venstre mus-tast. Da vil vinduet sprette frem og bli det øverste.

Vi kan oppnå det samme ved å peke og klikke på programmets navn i programoversikten nederst på skjermen.

4.1.5 Flytting av vinduer

Det er også mulig for brukeren å flytte et vindu til et annet sted på skjermen. Pek først på toppfeltet i det vinduet som skal flyttes. Trykk så på *venstre tast* på musen og hold mus-tasten nede. Nå skal du kunne flytte musen og se at en skygge av vinduet følger etter. Når du til slutt er fornøyd med den nye posisjonen, kan du bare slippe mus-tasten.

Denne teknikken kan også brukes til å flytte et vindu til en annen delskjerm; bare dra vinduet «ut av» skjermen, så dukker det opp på en annen delskjerm.

4.1.6 Endring av størrelsen på vinduer

Det er mulig å endre størrelsen på de fleste vinduene, og fremgangsmåten for dette er følgende:

- 1) Pek på det vinduet du ønsker å endre størrelsen på. Plasser mus-markøren oppå den sidekanten eller det hjørnet du ønsker å endre; skal du for eksempel gjøre vinduet høyere, skal mus-markøren plasseres på midten av nedre eller øvre kant. Når mus-markøren er riktig plassert, kan vi se at den skifter utseende for å markere hva den nå kan brukes til.
- 2) Trykk på *venstre mus-tast* og hold mus-tasten nede.
- 3) Nå kan du flytte musen og se at vinduet endrer størrelse, og ett eller annet sted på skjermen vil det stå angitt hvor stort vinduet er til enhver tid. Slipp mus-tasten når du er fornøyd.

4.1.7 Menyer

Det er mulig å benytte musen til å gi kommandoer til systemet ved hjelp av to ulike typer menyer.

4.1.7.1 Vindusmenyen

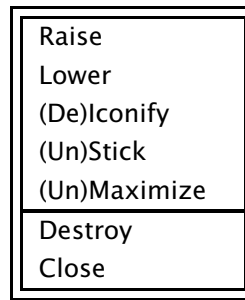
De fleste vinduer har sin egen meny over mulige operasjoner; stort sett har disse samme utseende slik det er vist i figur 4.1 på neste side. Denne menyen kan hentes frem ved å peke på menymerket merket «▽» til venstre i toppfeltet i vinduet og klikke med venstre mus-tast. De ulike feltene i vindusmenyen er:

Raise vil legge vinduet øverst i «bunken» av synlige vinduer.

Lower vil legge vinduet underst i «bunken» av synlige vinduer.

(De)Iconify vil ikonifisere bildet; du kan lese mer om *Ikoner* på neste side.

(Un)Stick vil «klistre» vinduet fast til skjermen; da vil man se det uansett hvilken delskjerm man titter på. Neste valg vil gjøre vinduet normalt igjen.



Figur 4.1: Vindusmenyen

(Un)Maximize vil gjøre vinduet så stort som mulig. Neste gang man velger dette, får man tilbake den opprinnelige størrelsen.

Destroy vil drepe programmet som styrer vinduet og fjerne vinduet for godt.

Close vil sende et signal til vinduets program om at dette vinduet ikke lenger er ønsket. Hvis programmet har flere vinduer, vil bare dette ene bli fjernet.

4.2 Ikoner

Hvis man etter hvert har opprettet mange vinduer, kan det bli ganske rotete på skjermen. Det er da mulig å omdanne vinduer til såkalte **ikoner**;³ da forsvinner vinduet, men navnet blir stående i programoversikten nederst på skjermen. Brukeren kan siden hente frem det opprinnelige vinduet.

4.2.1 Å lage ikoner

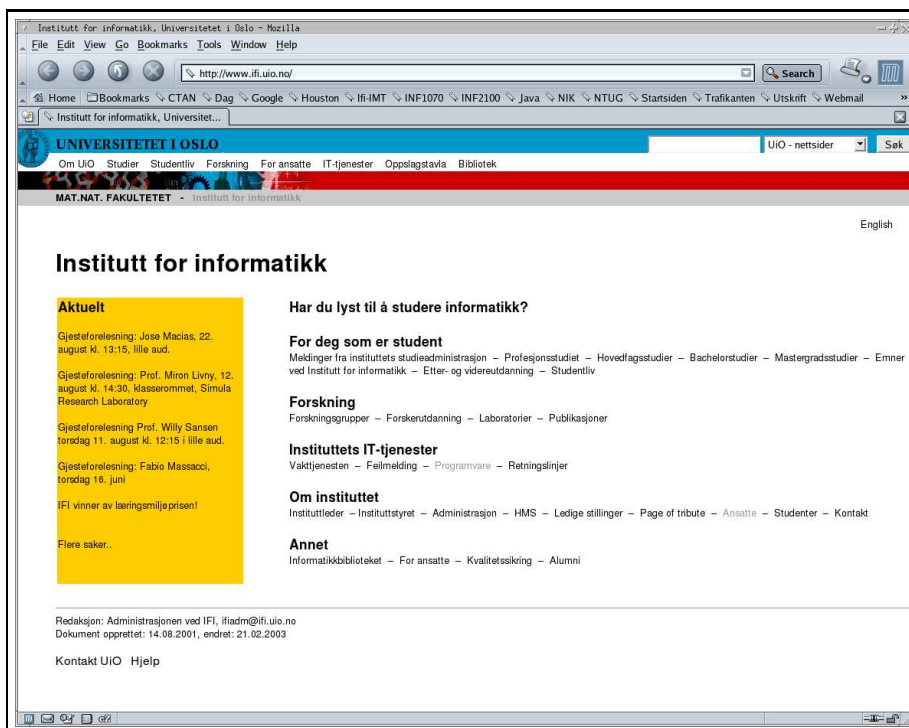
Det er to forskjellige måter å omdanne et vindu til sitt ikon på:

- enten peke på ikoniseringsmerket i toppfeltet i vinduet («–» plassert som merke nummer tre fra høyre) og klikke med venstre mus-tast, eller
- velge kommandoen (De)Iconify i vindusmenyen.

4.2.2 Å hente vinduer tilbake

Det er svært enkelt å omforme et ikon tilbake til sitt opprinnelige vindu; det er faktisk bare å klikke på programmets navn i programoversikten nederst på skjermen.

³ Hvis oppsett har en programoversikt nederst på skjermen, og programnavnet der er eneste spor etter det ikonifiserte programmet. Andre oppsett omformer i stedet vinduet til et lite bilde (derav navnet *ikon*) som plasseres i hjørne av skjermen.



Figur 4.2: VVV-leseren Mozilla

4.3 Vvv og programmet Mozilla

Verdensveven (vanligvis forkortet vvv⁴) er et system for utveksling av informasjon over Internett. Her finnes informasjon på ulike former som tekst, bilder, tegnefilmer og lyd, og alt er like lett tilgjengelig enten det ligger lagret her ved Ifi eller hvor som helst i verden. Omtrent all informasjon om instituttet og livet der er nå lagt opp i vvv.

Det finnes flere programmer som kan hente informasjon i vvv; her ved Ifi !
satser vi mest på Mozilla. Dette kan startes ved å klikke på ikonet



eller ved å gi mozilla som en vanlig UNIX-kommando. Da kommer det frem et vindu som vist i figur 4.2.

⁴ Verdensveven (forkortet vvv) er den norsk betegnelsen på «World wide web» (www).

Selv om dette er galskap, er det
likevel system i den.
— William Shakespeare

Kapittel 5

Filer

Dette kapitlet beskriver filsystemet under UNIX, og det vil ta for seg filområder, filstruktur, notasjon for filnavn samt beskyttelse av filer.

5.1 Informasjon om filer

Som nevnt tidligere (i avsnittet [Oversikt over filer](#) på side 9) benyttes kommandoen `ls` til å gi en oversikt over hvilke filer en bruker har. Denne kommandoen kan gis opsjoner slik at man får utskriften på den formen man ønsker. De viktigste opsjonene er:

`-l` (for «long») vil gi en fyldigere utskrift med informasjon om hver fil. Utskriften vil se slik ut:

```
hunding: dag> ls -l
total 204
-rw----- 1 dag      23 Jan  6 15:23 Fil1
-rw----- 1 dag     224 Jan  8 12:01 Fil2
```

Den består av følgende elementer:

- Det aller første tegnet ('-' i eksemplet) angir hva slags fil det er snakk om. De to vanligste formene er 'd' og '-':
 - d** angir at filen representerer et filområde (se avsnittet [Fil-områder](#) på neste side).
 - angir at det er snakk om en vanlig fil.
- Så følger 9 tegn som angir beskyttelseskoden for filen (denne koden er 'rw-----' i eksemplet). Koden for filbeskyttelse omtales nærmere i avsnittet [Fil-beskyttelse](#) på side 45.
- Deretter følger antall referanser til filen (1 i dette eksemplet). Vi skal ikke her komme nærmere inn på hva det innebærer.
- Nå kommer eierens navn (dag i dette tilfellet).
- Det neste tallet (23 og 224) angir størrelsen på filen, med andre ord hvor mange tegn¹ den inneholder.

¹ Egentlig angir tallet antall *byte* filen inneholder; byte er den basale lagringsenheten for UNIX-filer. For filer som inneholder lesbar tekst, vil dette imidlertid være det samme som antall tegn.

- Så kommer datoen og klokkeslettet da filen ble laget.
 - Til sist kommer filens navn.
- a (for «all») vil ta med alle filene i oversikten, også de hvis navn starter med '.'. Slike filer omtales nærmere i avsnittet *Usynlige filer* på side 50.
- F (for «flag») vil legge en markering om hva slags fil det er snakk om bak alle filnavnene den skriver ut. De vanligste markeringene er:
- / markerer at filen representerer et filområde.
 - * angir at filen er eksekverbar, det vil si at den kan utføres som en kommando.
 - @ viser at filen er en såkalt *link* (se avsnittet *Linker* på side 49).
- Vanlige filer får ingen markering.
- g (for «group») angir at også filens *gruppe* skal vises. Grupper omtales i avsnittet *Fil-grupper* på side 47.
- t (for «time») vil sørge for at listen blir sortert etter når filen ble laget i stedet for alfabetisk etter navn som er det vanlige.

5.2 Fil-områder

Etter hvert som man arbeider mot datamaskinen, vil man lage flere og flere filer, og etter kort tid vil det bli vanskelig å holde oversikten. For å hjelpe på dette, har brukere anledning til å lage *filområder*, også kalt *kataloger* eller *mapper* (eller «directories» eller «folders» på engelsk).

Et filområde er en spesiell fil som kan inneholde andre filer. Nye brukere har bare ett filområde (kalt *hovedområdet* eller *hjemmeområdet*) hvor alle deres filer ligger, men de har muligheten til å lage nye områder for filer som hører logisk sammen.

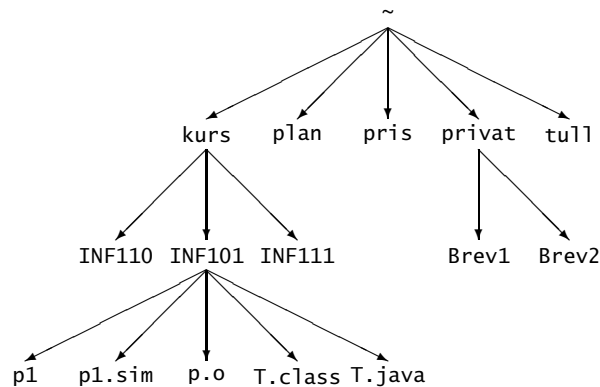
Siden et filområde betraktes som en slags fil av systemet, er det fullt mulig at et filområde kan inneholde et annet filområde, som igjen kan inneholde et nytt, og så videre. Dette gir en struktur som kalles et *tre*, og et typisk tre kan beskues i figur 5.1 på neste side.²

5.2.1 Notasjon for filer og fil-områder

Så lenge det bare var snakk om ett filområde (hovedområdet), var det intet problem å angi hvilken fil man mente; man bare oppga navnet. Når brukeren oppretter filområder, er det imidlertid verre å vite hvilken fil man ønsker; det er nemlig tillatt at filer i ulike filområder kan ha samme navn.

Når man skal angi et filnavn et annet sted enn hovedområdet, må man angi veien fra hovedområdet og ned til det. Hvis vi antar at filstrukturen er som vist i figur 5.1 på neste side, vil vi ha følgende notasjon for filene:

² Av en eller annen merkelig grunn liker databehandlere å tegne trær opp ned. Undres hva Freud ville sagt om det...



Figur 5.1: Et typisk tre av brukerfiler

- plan kan angis uten noen videre dikkedarer, siden den ligger i hovedområdet.
- privat/Brev2 ligger i området privat, så det må med i notasjonen.
- kurs/INF101/p1.sim ligger i området INF101 som igjen er et område under kurs.

Det finnes også andre notasjoner; vi kommer tilbake til dem senere (nærmere bestemt i avsnittet [Mer om notasjon for fil-navn](#) på side 43 og i [Notasjon for filer i systemet](#) på side 45).

5.2.2 Opprettelse av fil-områder

Nye filområder opprettes med kommandoen `mkdir`.³ Anta at vi på hovedområdet kun har 2 filer:

```
hunding: dag> ls -F
plan  tull
```

Så oppretter vi et nytt område som vi ønsker skal hete kurs (sannsynligvis fordi det skal inneholder filer som har noe med kurs å gjøre).

```
hunding: dag> mkdir kurs
```

Da kan vi se at det er oppstått et nytt område:

```
hunding: dag> ls -F
kurs/  plan  tull
```

(Skråstreken bak navnet viser at det er snakk om et område og ikke en vanlig fil. Dette merket dukket opp fordi vi benyttet opsjonen '-F' til `ls`.)

5.2.3 Flytting av filer mellom filområder

Hvis vi ønsker å rydde opp i filene våre, kan vi for eksempel flytte filen `plan` over i området `kurs`:

³ Navnet `mkdir` er en forkortelse for *make directory*.

```
hunding: dag> move plan kurs/plan
```

Kommandoen `move` (se avsnittet [Endring av navn på filer](#) på side 24) benyttes til å flytte filer, enten innen et område eller mellom områder. En omnavning betraktes som en flytting av filen innen samme område.

Hvis man ønsker at filen skal beholde sitt navn når den flyttes til et annet område, er det nok å angi området filen skal flyttes til. Eksemplet over kunne dermed også vært skrevet:

```
hunding: dag> move plan kurs
```

Dette gjelder også for kopieringskommandoen `copy` (se avsnittet [Kopiering av filer](#) på side 23).

5.2.4 Fjerning av fil-områder

Hvis man ikke lenger trenger et område, kan det fjernes med kommandoen `rmdir`.⁴ Av sikkerhetsgrunner må området være tomt før det kan fjernes, ellers kunne man av vanvare ødelegge mange filer. I dette tilfellet ville man heller ikke ha noen angremulighet.

5.2.5 Manøvrering i fil-områdene

Som bruker kan man altså bygge opp en trestruktur av filområder. For å gjøre det enkelt å snakke om filer på et vilkårlig område, har operativsystemet UNIX bygget inn muligheten for å «flytte» seg rundt i strukturen.

5.2.5.1 I hvilket fil-område er jeg?

Kommandoen `pwd`⁵ skriver ut hvilket område jeg for øyeblikket befinner meg i:

```
hunding: dag> pwd  
/home/hunding/a/dag
```

(Dette er hovedområdet til brukeren dag. Mer om dette i avsnittet [Operativsystemets fil-struktur](#) på side 44.)

5.2.5.2 Å flytte til nytt område

Kommandoen `cd`⁶ benyttes til å flytte seg til et vilkårlig annet filområde. I sin enkleste form, det vil si uten parametre, vil `cd` flytte til brukerens hovedområde:

```
hunding: dag> cd  
hunding: dag> pwd  
/home/hunding/a/dag
```

⁴ Navnet `rmdir` er en forkortelse for *remove directory*.

⁵ Navnet `pwd` er en forkortelse for *print working directory*.

⁶ Navnet `cd` står for *change directory*.

Posisjon i treet	Notasjon for filen p1.sim
~	kurs/INF101/p1.sim
~/kurs	INF101/p1.sim
~/kurs/INF101	p1.sim
~/kurs/INF111	../INF101/p1.sim
~/privat	../kurs/INF101/p1.sim

Tabell 5.1: Eksempler på bruk av relativ notasjon for fil-navn

Hvis vi angir et filområde som parameter, vil `cd` flytte oss dit. Anta at vi på hovedområdet har et filområde som heter `kurs` og ønsker å flytte oss dit.

```
hunding: dag> pwd
/home/hunding/a/dag
hunding: dag> ls -F
kurs/ plan tull
hunding: dag> cd kurs
hunding: kurs> pwd
/home/hunding/a/dag/kurs
```

Vi ser her at klarsignalet endrer seg etter `cd`-kommandoen. Klarsignalet viser alltid siste ledd av området vi befinner oss i.

5.2.6 Mer om notasjon for fil-navn

Forrige avsnitt om notasjon for filnavn tok utgangspunkt i at brukeren alltid befant seg i sitt hovedområde. Det er jo ikke alltid tilfelle, så måten å angi filnavn på må revurderes.

Det kan benyttes to forskjellige notasjoner for filnavn:

- **Absolutt notasjon** starter alltid med et fast utgangspunkt, nemlig brukerens hovedområde, og angir veien derfra ned til filen. Tegnet '~' benyttes for å angi brukerens hovedområde. For å finne frem til filen `p1.sim` i figur 5.1 på side 41, er den absolutte notasjonen

```
~/kurs/INF101/p1.sim
```

Den absolutte notasjonen for en fil er alltid den samme uansett hvor brukeren har flyttet seg i filstrukturen sin.

- **Relativ notasjon** tar utgangspunkt i hvor i filstrukturen brukeren har plassert seg, og følger veien derfra. Følgelig vil notasjonen for én og samme fil variere med hvor brukeren er plassert, som vist i tabell 5.1. (Notasjonen '..' er forklart i neste avsnitt.)

5.2.6.1 Spesielle tegn i notasjonen for fil-navn

Noen tegn har en spesiell betydning når de forekommer i notasjonen for et filnavn. Disse tegnene er:

~ brukes, som tidligere nevnt, til å angi brukerens hovedområde. Det er altså en absolutt angivelse.

- . benyttes til å angi «området selv». Den vanligste bruken er i forbindelse med flytting av filer. Ønsker jeg for eksempel å flytte filen `obl-1.sim` fra området `INF101` til det området jeg befinner meg i, kan dette skrives enkelt som

```
hunding: dag> move INF101/obl-1.sim .
```

- .. benyttes til å angi «området over» (ofte kalt *foreldreområdet*). Et eksempel vil kanskje gjøre dette klarere:

```
hunding: INF101> pwd  
/home/hunding/a/dag/kurs/INF101  
hunding: INF101> cd ..  
hunding: kurs> pwd  
/home/hunding/a/dag/kurs
```

Brukeren var plassert i området `INF101`, og ba om å bli flyttet til `..`, altså ett hakk opp. Dermed ble brukeren plassert i området `kurs` som er filområdet «over».

5.3 Joker-notasjon for fil-navn

Av og til kan man ønske å omtale mange filnavn samtidig. Et eksempel er at man ønsker en oversikt fra `ls` over alle filer på området hvis navn ender på `.java`.

Dette ordnes enkelt i UNIX med det som kalles *jokernotasjon* (på engelsk «wild card notation»), det vil si en notasjon som angir at «hva som helst» passer. Til dette benyttes to spesielle tegn:

- ? angir at et hvilket som helst enkelttegn passer. Et eksempel på dette er:

```
hunding: dag> ls  
p1.java p2.java p3.java prog1.java prog23.java  
hunding: dag> ls p?.java  
p1.java p2.java p3.java
```

De tre første filnavnene passet, men ikke de to siste, for der var forskjellen mer enn ett enkelttegn.

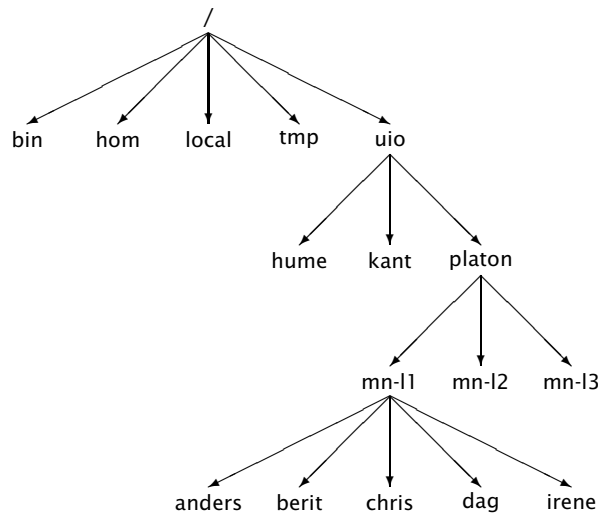
- * angir at et vilkårlig antall tegn passer, som følgende eksempel skulle vise:

```
hunding: dag> ls  
p1.java p2.java p3.java prog1.java prog23.java  
hunding: dag> ls prog*.java  
prog1.java prog23.java  
hunding: dag> ls *.java  
p1.java p2.java p3.java prog1.java prog23.java
```

5.4 Operativsystemets fil-struktur

Som nevnt kan hver bruker bygge seg sin egen filstruktur i form av et tre hvor brukerens hovedområde er roten. Dette er imidlertid bare en del av virkeligheten. Faktisk er den enkelte brukers filstruktur bare en liten kvist av den store trestrukturen som omfatter alle filer i alle UNIX-datamaskiner ved Universitetet. Et utsnitt av strukturen er vist i figur 5.2 på neste side.⁷ Roten i systemets struktur har det merkelige navnet `/`.

⁷ I figuren er det vist en *logisk struktur*, altså slik den ser ut for brukeren. Den *fysiske struktur* vil være noe anderledes, blant annet fordi filene ligger på diskene på ulike maskiner. Dette



Figur 5.2: Operativsystemets filstruktur

5.4.1 Notasjon for filer i systemet

Når man skal spesifisere hvilken fil i hele filsystemet man mener, benyttes samme teknikk som for filer i ett av ens egne filområder. Forskjellen er at man i tillegg kan benytte et par andre startpunkter for absolutt notasjon:

- Notasjonen '/' angir roten i hele filsystemet. Et eksempel på en absolutt filangivelse som tar utgangspunkt i roten, er

```
/uio/platon/mn-l1/dag/kurs/INF101/p1.sim
```

- Notasjonen '~*bruker*' angir en brukers hovedområde. En fil i bruker chris' hovedområde kan man således få tak i ved å skrive

```
~chris/fil
```

- Notasjonen '/*hom/bruker*' er en alternativ måte å angi ~*bruker* på. Dette er av og til nødvendig fordi ikke alle programmer godtar '~'-notasjonen.⁸

5.5 Fil-beskyttelse

UNIX har et ganske enkelt system for beskyttelse av filer, men det er tilstrekkelig i de fleste tilfeller.

5.5.1 Beskyttelse av filer

Grunnlaget for filbeskyttelsen er at det for hver fil angis hvilke *privilegier* ulike brukere har overfor filen. Det er snakk om tre former for privilegier:

r angir *leseprivilegium*. Kun de som har r-privilegium kan lese en fil.

trenger ikke en vanlig bruker bekymre seg om.

⁸ Området /hom er ikke standard i UNIX; det er laget spesielt her ved Ifi.

w angir *skriveprivilegium*. Kun de som har **w**-privilegium kan skrive på filen.

x angir *eksekveringsprivilegium*. Kun de som har **x**-privilegium kan utføre et program.

UNIX regner også med at alle brukerne av datamaskinen kan deles inn i tre kategorier:

u (for «user») er brukeren selv, altså *eieren* av filen, den som laget den.

g (for «group») er andre brukere i samme *gruppe*. Grupper omtales nærmere i avsnittet *Fil-grupper* på neste side.

o (for «other») er alle andre.

Når man får oppgitt beskyttelseskoden for en fil slik det omtalt i avsnittet *Informasjon om filer* på side 39, kommer den i form av 3×3 tegn:

$$\begin{array}{ccc} \text{u} & \text{g} & \text{o} \\ \hline \text{rwx} & \text{r-x} & \text{r--} \end{array}$$

De første 3 angir privilegiene som gjelder for brukeren selv ('u'), de neste 3 er for gruppen ('g'), og de siste 3 for alle andre ('o'). For hver brukerkategori angis hvilke privilegier folk i den kategorien har, og de angis alltid i rekkefølgen rwx. Hvis brukere i kategorien har privilegiet, vises bokstaven, mens en '-' angir at de ikke har privilegiet.

I eksemplet har eieren av filen alle privilegier, gruppen har lese- og eksekveringsprivilegier mens alle andre kun har leseprivilegium.

5.5.1.1 Beskyttelse av fil-områder

Filområder benytter den samme formen for beskyttelse som enkeltfiler, men privilegiene tolkes litt forskjellig.

r angir *leseprivilegium*, det vil si mulighet til å se navnene på de filene som finnes på området. Dette innebærer ikke rett til å se på selve filene, til det kreves **x**-privilegium.

w angir *skriveprivilegium*, det vil si mulighet til å opprette nye filer på området og slette gamle.

x angir *søkeprivilegium*, det vil si mulighet til å se på filer på området.

Hvis man skal gi andre rett til å se på sine filer, er det altså ikke nok å markere filen åpen for lesing. Man må i tillegg sørge for at området er åpent for søking.

5.5.2 Endring av fil-beskyttelse

Kommandoen `chmod`⁹ benyttes til å endre beskyttelsen av filer og filområder. Anta at vi har følgende filer:

⁹ Navnet `chmod` er en forkortelse for *change mode*.

```
hunding: dag> ls -l
total 3
-rw-rw-r-- 1 dag          5 Jan 11 09:58 A
-rw----- 1 dag          5 Jan 11 09:58 B
-rw----- 1 dag          5 Jan 11 09:58 C
```

Vi ønsker å gjøre følgende endringer:

- 1) Ingen andre enn jeg selv skal ha noen privilegier for A.
- 2) Alle skal få lese C.

Kommandoene for å oppnå dette er:

```
hunding: dag> chmod go-rw A
hunding: dag> chmod go+r C
```

Første parameter til `chmod` angir hvilke endringer av privilegier som skal utføres. Dette er et treleddet uttrykk:

- Først angis hvilke kategorier brukere det er snakk om. I eksemplet her dreide det seg om 'g' og 'o', altså alle andre enn eieren.

Her kan man også bruke bokstaven 'a' (for «all») for å betegne alle brukere. Det siste eksemplet over kunne da vært skrevet

```
hunding: dag> chmod a+r C
```

(Hva ville skjedd hvis vi hadde benyttet en 'a' i det første eksemplet og skrevet `chmod a-rw A`? Da hadde vi ikke fått lov til å lese eller skrive vår egen fil! Vi ville imidlertid fått lov til å endre beskyttelsen igjen.)

- Så følger en '+' hvis det er snakk om privilegier noen skal få, eller en '-' hvis privilegier skal inndras.
- Til sist kommer hvilke privilegier det er snakk om.

Etter denne spesielle første parameteren følger listen av filer hvis privilegier skal endres.

Etter disse endringene vil filene ha følgende beskyttelse:

```
hunding: dag> ls -l
total 3
-rw----- 1 dag          5 Jan 11 09:58 A
-rw----- 1 dag          5 Jan 11 09:58 B
-rw-r--r-- 1 dag          5 Jan 11 09:58 C
```

5.5.3 Fil-grupper

For å kunne gi spesielle filprivilegier til et mindre antall brukere, er det mulig å definere **grupper** av brukere. Ved å gi kommandoen `groups` kan man se hvilke grupper man er med i:

```
hunding: dag> groups
dag olaf
```

Jeg er i dette eksemplet med i min egen gruppe (med samme navn som brukernavnet mitt) og gruppen 'olaf' som tilhører en kollega.

5.5.3.1 Å opprette nye grupper

! Alle Ifi-brukere får lov til å opprette sin egen gruppe, og de kan melde hvem de vil inn i den gruppen. (Dette er den eneste gruppen vanlige brukere kan melde noen inn i.) Kommandoen `bofh`¹⁰ benyttes til alt som har med styring av grupper å gjøre:

```
hunding: dag> bofh
Bofhd server is at https://cerebrum.uio.no:8000
Password for dag: xxxxxxxx
Welcome to jbofh, v 0.9.5, type ‘‘help’’ for help
jbofh> group personal
Enter account name >dag
jbofh> quit
I’ll be back
```

Nå er altså min egen brukergruppe `dag` opprettet; den har alltid samme navn som mitt brukernavn.

5.5.3.2 Å melde folk inn i en gruppe

Jeg kan nå bruke `bofh` til å legge mine venner og samarbeidspartnere inn i gruppen min:

```
jbofh> group add peterax dag
OK, added peterax to dag
```

Nå er brukeren `peterax` også medlem i gruppen `dag`, som er *min* gruppe. Det kan ta et par timer før opplysningene om gruppeendringen er nådd ut til alle maskinene; dessuten må brukere logge ut og inn igjen før systemet registrerer at det er skjedd en endring.

Hvis man glemmer hvem som er med i ens gruppe, kan `bofh` fortelle det:

```
jbofh> group list dag
MemberOp  Type      Name
union      account   dag
union      account   olaf
union      account   peterax
```

5.5.3.3 Endring av fil-gruppe

Kommandoen `chgrp`¹¹ benyttes for å angi at en fil skal tilhøre en spesiell gruppe:

```
hunding: dag> chgrp dag fil22.txt
hunding: dag> chmod g+r fil22.txt
hunding: dag> chmod o-r fil22.txt
hunding: dag> ls -l -g fil22.txt
-rw-r----- 1 dag dag 26588 Jul 2 14:19 fil22.txt
```

Når har filen `fil22.txt` gruppe `dag` og den er lesbar for gruppen men ikke for andre. Dermed kan `peterax` få se på den siden hun nå er med i gruppen `dag`, men ingen andre.

¹⁰ Navnet `bofh` er en forkortelse for det litt eiendommelige navnet *brukerorganisering for hvermann*.

¹¹ Navnet `chgrp` er en forkortelse for *change group*.

umask-kode	Enkelt-filer	Fil-områder
77	rw- --- ---	rwX --- ---
27	rw- r-- ---	rwX r-x ---
22	rw- r-- r--	rwX r-x r-x
2	rw- rw- r--	rwX rwX r-x
0	rw- rw- rw-	rwX rwX rwX

Tabell 5.2: Vanlige umask-koder

Man får bare lov å endre gruppen på filer man selv eier, altså filer man selv har laget, og man får bare lov å endre gruppen til en gruppe man selv er med i.

5.5.4 Standard fil-beskyttelse

Når man lager nye filer eller nye filområder, vil de få en standard beskyttelseskode. Man kan benytte kommandoen `umask` til å spørre hva denne koden er:

```
hunding: dag> umask
77
```

Dessverre blir denne koden oppgitt som et kryptisk tall, men ved hjelp av tabell 5.2 kan man finne sammenhengen mellom de vanligste tallverdiene og den beskyttelseskode de representerer for henholdsvis enkeltfiler og filområder.

Det er også mulig å benytte `umask`-kommandoen til å endre standard filbeskyttelse:

```
hunding: dag> umask 22
```

Heretter vil alle nye filer være lesbare for alle.¹²

En slik endring av standard filbeskyttelse gjelder inntil man endrer den på nytt eller man logger ut. Hvis man ønsker en mer permanent endring, kan man legge kallet på `umask` inn i sin `.envir`-fil, slik det står omtalt i avsnittet [Inn- og utlogging](#) på side 64.

5.6 Linker

I UNIX er det mulig å opprette **linker** til filer. En link er en spesialfil som er en referanse til en annen fil. Anta at vi har en fil på det området vi har flyttet oss til:

¹² Som du kanskje har lagt merke til, finnes det ingen `umask`-kode som automatisk gjør nye filer eksekverbare. Det må gjøres eksplisitt av brukeren med kommandoen `chmod`. Spesielle programmer, som for eksempel C-kompilatoren, vet at de lager en eksekverbar fil, og de vil selv sette eksekveringsprivilegiet.

```
hunding: pris> ls
geitost
hunding: pris> more geitost
Geitost koster kr 65,50 hos Rema 1000
kr 67,45 hos Rimi
```

Vi kan da lage en link til geitost med kommandoen `ln` og opsjonen `-s`¹³:

```
hunding: pris> ln -s geitost ost
hunding: pris> ls -F
geitost  ost@
```

Vi har da to navn på samme fil: det opprinnelige `geitost` og linken `ost`. For alle praktiske formål er det det samme hvilken av de to vi bruker:

```
hunding: pris> more ost
Geitost koster kr 65,50 hos Rema 1000
kr 67,45 hos Rimi
```

Det som gjør linker spesielt nyttige, er at de kan referere filer et helt annet sted i filtreet. Hvis jeg for eksempel ofte titter på innføringsheftet *L^AT_EX for nybegynnere*, kan jeg lage en link til den aktuelle filen:

```
hunding: pris> ln -s /store/share/doc/ifi/LaTeX/LaTeX-for-nybegynnere.pdf ~/lfn.pdf
```

Siden dette er en referanse til filen og ingen kopi, vil jeg alltid få siste versjon av filen når jeg skriver `preview ~/lfn.pdf`.

5.7 Spesielle filer

Enkelte filer har en spesiell funksjon under UNIX, og dette markeres ofte ved at de har et spesielt navn. I dette avsnittet skal vi se på de vanligste variantene.

5.7.1 Usynlige filer

Hos enhver bruker finnes det filer som man vanligvis ikke er interessert i fordi de inneholder spesifikasjoner om oppsett og lignende som sjelden eller aldri endres. Ofte er dette filer som andre, for eksempel driftspersonalet, har laget. Slike filer vil man helst ikke ha med i filoversikter fra `ls`.

I UNIX har man en fast konvensjon om at slike filer skal ha et navn som begynner med `'.'`. Kommandoen `ls` tar hensyn til dette, og tar ikke med slike usynlige filer, med mindre brukeren spesielt ber om det ved å benytte opsjonen `-a`:

```
hunding: dag> ls
kurs plan
hunding: dag> ls -a
.      .rwmjobs      .emacs      .newsrsrc
..     .rwmrc       .login      kurs
.Xlog  .cshrc         .logout     plan
```

¹³ Ved å ta med opsjonen `-s` får vi opprettet en såkalt *symbolsk link* til filen. Dropper vi opsjonen, får vi en *hard link* og det ønsker vi ikke, uten at jeg skal komme nærmere inn på forskjellen.

Endelse	Filen inneholder
.c	Et C-program.
.cc	Et program i C++.
.class	Kode generert fra et Java-program
.dvi	Resultatet av en T _E X- eller L ^A T _E X-kjøring. ¹⁵
.gz	En fil som er komprimert med gzip.
.h	En definisjonsfil for et C-program.
.java	Et JAVA-program
.ps	En POSTSCRIPT-fil. ¹⁶
.tex	Et dokument i T _E X eller L ^A T _E X. ¹⁷
.Z	En fil som er komprimert med compress.

Tabell 5.3: Vanlige endelser for filnavn

5.7.2 Midlertidige filer

Av og til ønsker man å lage en arbeidsfil som man trenger noen minutter og deretter bare skal fjerne. I UNIX er det vanlig å la navnet på slike filer starte med en '#'. Slike filer vil automatisk bli fjernet av systemet etter 3 døgn. Det er godt å vite hvis man selv skulle glemme å fjerne dem.

5.7.3 Filer med spesielt innhold

Det er vanlig i UNIX at navnet på en fil forteller litt om hva den inneholder. Dette gjøres ved å benytte spesielle endelser.¹⁴ De vanligste her ved instituttet er gitt i tabell 5.3. Det er vanlig at eksekverbare programmer ikke har noen endelse i det hele tatt.

5.7.4 Spesielle fil-navn

Enkelte filnavn har en fast betydning i UNIX:

core er navnet på en fil med kopi av hele lagerinnholdet (en såkalt *dump*). Hvis et program inneholder feil, hender det at ting stopper helt opp. Da vil programmet lage en fil som inneholder hele arbeidslageret slik at dette kan analyseres for å finne feilen. Denne filen heter *core*.¹⁸

¹⁴ Her har UNIX en annen filosofi enn mange andre operativsystemer som for eksempel WINDOWS. Filnavn i UNIX har ingen struktur med *hoveddel* og *endelse*; de består rett og slett av en sekvens tegn, og det er opp til det enkelte program om det vil legge noen betydning i de siste tegnene av filnavnet.

¹⁵ Slike filer (hvis navn slutter på '.dvi') er meget spesielle; de kan ikke skrives ut på skjermen med more. I stedet må man benytte programmet *preview* (se avsnittet [Utskrift av filer på skjermen](#) på side 22) for å se innholdet.

¹⁶ POSTSCRIPT-filer trenger også et spesielt program for å kunne vises på skjermen; bruk *preview* (se avsnittet [Utskrift av filer på skjermen](#) på side 22).

¹⁷ T_EX og L^AT_EX er ypperlige hjelpemidler for å skrive pene og oversiktlige dokumenter. Dette kompendiet er for eksempel skrevet med L^AT_EX. For mer informasjon om L^AT_EX, les heftet *L^AT_EX for nybegynnere*.

¹⁸ Navnet *core* skyldes at maskiner i gamle dager (det vil si før 1975) hadde hurtiglager bygget opp av små magnetiske ferittringer kalt *kjerner*. En *core dump* var altså opprinnelig en utskrift at alt som var lagret i kjernene.

Siden de færreste brukere ved Ifi er interessert i denne filen, er muligheten for å lage en core-fil normalt slått av.

5.8 Automatisk fjerning av filer

For å unngå at uviktige filer ikke blir liggende og oppta verdifull plass, er det innført en ordning hvor følgende filer blir fjernet hvis de er eldre enn 3 dager:

core #* *.log *.dvi *.CKP *.BAK *~

(Legg merke til at noen av filene her er angitt med jokernotasjone; se avsnittet *[Joker-notasjon for fil-navn](#)* på side [44](#).)

Kapittel 6

For viderekomne

Dette kapittelet tar for seg en del finesser ved UNIX man bør vite om når man er over det første begynnerstadiet.

6.1 Tegnsett

Ifis maskiner benytter *ISO 8859-1* (også kjent som *ISO Latin-1*) som standard tegnsett. Dette tegnsettet er en utvidelse av det kjente ASCII og ser ut som vist i tabell 6.1 på neste side.

6.2 Gjenbruk av tidligere kommandoer

Det finnes muligheter til å hente frem tidligere kommandoer, rette dem og så få dem utført. Til dette benyttes piltastene. Opp- og ned-pilene blar gjennom listen over tidligere kommandoer; opp-pilen blar bakover i listen og ned-pilen fremover.

Hvis man på denne måten finner en kommando man vil gjenta, er det bare å trykke . Alternativt kan man rette litt på den før man sender den til utførelse. Med venstre- og høyre-pilene kan man flytte seg innen kommandoen. Selve redigeringen skjer med de samme kontrolltastene som benyttes i Emacs; de viktigste er vist i tabell 6.2 på side 55.

6.3 Noen nyttige programmer

6.3.1 Programmet w

Programmet w¹ forteller hvem som er logget inn på den datamaskinen man kjører mot, og hva de gjør.² Når man er logget inn på en selvstendig arbeidsstasjon, er dette sjelden interessant, fordi man da gjerne er alene på maskinen. Kjører man derimot på en sentral tjenermaskin («server» på engelsk), kan man imidlertid ønske å se hvilke andre brukere som man deler maskinen med og som sannsynligvis har skylden for at maskinen går så tregt.

¹ Navnet w er en forkortelse for *what*.

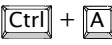
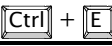
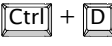
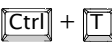
² Dette er et typisk eksempel på hvor åpent UNIX er; systemet regner med at ingen har noen grunn til å skjule for andre hva de driver med.

ISO 8859-1

0	000	32	040	64	@	100	96	‘	140	128	200	160	240	192	À	300	224	à	340
	00		20			40			60		80		A0			C0		è	E0
1	001	33	041	65	A	101	97	a	141	129	201	161	241	193	Á	301	225	á	341
	01		21			41			61		81		A1			C1		é	E1
2	002	34	042	66	B	102	98	b	142	130	202	162	242	194	Â	302	226	â	342
	02		22			42			62		82		A2			C2		ê	E2
3	003	35	043	67	C	103	99	c	143	131	203	163	243	195	Ã	303	227	ã	343
	03		23			43			63		83		A3			C3		ë	E3
4	004	36	044	68	D	104	100	d	144	132	204	164	244	196	Ä	304	228	ä	344
	04		24			44			64		84		A4			C4		ö	E4
5	005	37	045	69	E	105	101	e	145	133	205	165	245	197	Å	305	229	å	345
	05		25			45			65		85		A5			C5			E5
6	006	38	046	70	F	106	102	f	146	134	206	166	246	198	Æ	306	230	æ	346
	06		26			46			66		86		A6			C6			E6
7	007	39	047	71	G	107	103	g	147	135	207	167	247	199	Ç	307	231	ç	347
	07		27			47			67		87		A7			C7			E7
8	010	40	050	72	H	110	104	h	150	136	210	168	250	200	È	310	232	è	350
	08		28			48			68		88		A8			C8			E8
9	011	41	051	73	I	111	105	i	151	137	211	169	251	201	É	311	233	é	351
	09		29			49			69		89		A9			C9			E9
10	012	42	052	74	J	112	106	j	152	138	212	170	252	202	Ê	312	234	ê	352
	0A		2A			4A			6A		8A		AA			CA			EA
11	013	43	053	75	K	113	107	k	153	139	213	171	253	203	Ë	313	235	ë	353
	0B		2B			4B			6B		8B		AB			CB			EB
12	014	44	054	76	L	114	108	l	154	140	214	172	254	204	Ì	314	236	ì	354
	0C		2C			4C			6C		8C		AC			CC			EC
13	015	45	055	77	M	115	109	m	155	141	215	173	255	205	Í	315	237	í	355
	0D		2D			4D			6D		8D		AD			CD			ED
14	016	46	056	78	N	116	110	n	156	142	216	174	256	206	Î	316	238	î	356
	0E		2E			4E			6E		8E		AE			CE			EE
15	017	47	057	79	O	117	111	o	157	143	217	175	257	207	Ï	317	239	ï	357
	0F		2F			4F			6F		8F		AF			CF			EF
16	020	48	060	80	P	120	112	p	160	144	220	176	260	208	Ð	320	240	ð	360
	10		30			50			70		90		B0			D0			F0
17	021	49	061	81	Q	121	113	q	161	145	221	177	261	209	Ñ	321	241	ñ	361
	11		31			51			71		91		B1			D1			F1
18	022	50	062	82	R	122	114	r	162	146	222	178	262	210	Ò	322	242	ò	362
	12		32			52			72		92		B2			D2			F2
19	023	51	063	83	S	123	115	s	163	147	223	179	263	211	Ó	323	243	ó	363
	13		33			53			73		93		B3			D3			F3
20	024	52	064	84	T	124	116	t	164	148	224	180	264	212	Ô	324	244	ô	364
	14		34			54			74		94		B4			D4			F4
21	025	53	065	85	U	125	117	u	165	149	225	181	265	213	Õ	325	245	õ	365
	15		35			55			75		95		B5			D5			F5
22	026	54	066	86	V	126	118	v	166	150	226	182	266	214	Ö	326	246	ö	366
	16		36			56			76		96		B6			D6			F6
23	027	55	067	87	W	127	119	w	167	151	227	183	267	215	×	327	247	÷	367
	17		37			57			77		97		B7			D7			F7
24	030	56	070	88	X	130	120	x	170	152	230	184	270	216	Ø	330	248	ø	370
	18		38			58			78		98		B8			D8			F8
25	031	57	071	89	Y	131	121	y	171	153	231	185	271	217	Ù	331	249	ù	371
	19		39			59			79		99		B9			D9			F9
26	032	58	072	90	Z	132	122	z	172	154	232	186	272	218	Ú	332	250	ú	372
	1A		3A			5A			7A		9A		BA			DA			FA
27	033	59	073	91	[	133	123	{	173	155	233	187	273	219	Û	333	251	û	373
	1B		3B			5B			7B		9B		BB			DB			FB
28	034	60	074	92	\	134	124		174	156	234	188	274	220	Ü	334	252	ü	374
	1C		3C			5C			7C		9C		BC			DC			FC
29	035	61	075	93	]	135	125	}	175	157	235	189	275	221	Ý	335	253	ý	375
	1D		3D			5D			7D		9D		BD			DD			FD
30	036	62	076	94	^	136	126	~	176	158	236	190	276	222	Þ	336	254	þ	376
	1E		3E			5E			7E		9E		BE			DE			FE
31	037	63	077	95	_	137	127		177	159	237	191	277	223	ß	337	255	ÿ	377
	1F		3F			5F			7F		9F		BF			DF			FF

© April 1995, DFL, Iu/UGO

Tabell 6.1: Tegnsettet ISO 8859-1

Taster	Effekt
Flytting	
	Flytt ett ord bakover
	Flytt ett ord forover
	Flytt til starten av linjen
	Flytt til slutten av linjen
Sletting	
	Fjern tegnet til venstre for markøren
	Fjern tegnet til venstre for markøren
	Fjern tegnet under markøren
	Fjern forrige ord
	Fjern neste ord
Ombytting	
	Bytt om dette tegnet og det forrige
	Bytt om dette ordet og det forrige

Tabell 6.2: Kontrolltaster for redigering av kommandoer

```

hunding: dag> w
2:19pm up 16 hrs, 6 users, load average: 1.21, 1.36, 1.07
User      tty      login@   idle   JCPU   PCPU   what
fosli     tty00    11:34am   8      34     33    emacs
siri      tty01    1:22pm    0      20     20    emacs k7.tex
istvan    tty02    1:16pm   17      7      1    -csh
evy       tty0e    1:09pm   50      7      3    emacs IN324
arthur    tty0     2:14pm    3      1      1    mail
f-oil     tty7     1:44pm   35      2      1    -csh

```

Den første linjen forteller hva klokken er (14.19), hvor lenge siden maskinen ble startet sist (16 timer), hvor mange som for tiden er logget inn (6 brukere), og hvilken belastning det er på maskinen for øyeblikket (1,21 siste minutt, 1,36 siste 5 minutter og 1,07 siste kvarter). Så følger en overskrift og deretter én linje for hver bruker som er logget inn; de viktigste opplysningen her er brukernavnet ('User') og hva de gjør ('what'). Ytterligere opplysninger om kommandoen w kan man få ved å skrive man w.

6.3.2 Programmet photo

Ofte ønsker man å få en logg på papir av det arbeidet man gjør, med de kommandoer man gir maskinen og de svar maskinen gir tilbake. Dette oppnås enkelt med kommandoen photo hvor man oppgir navnet på en fil hvor denne loggen skal lagres:

```

hunding: dag> photo nylogg.log
Starting photo. Type 'exit' to quit photo

```

```
Script started, file is nylogg.log
hunding: dag>
```

Man starter da en sesjon med en ny kommandotolker som varer inntil man gir kommandoen exit:

```
hunding: dag> exit
exit
Script done, file is nylogg.log
hunding: dag> ls -l nylogg.log
-rw-r--r-- 1 dag          196 Jul  6 18:04 nylogg.log
```

Nå er sesjonen over og vi er tilbake der vi var, men med en logg-fil over det vi har gjort.³

6.3.3 Programmet tail

Ofte ønsker man å se hva som ligger på slutten av en fil uten å måtte skrive ut hele filen. Til dette benyttes kommandoen tail:

```
hunding: dag> tail -2 hva.txt
arthur  tty0    2:14pm    3      1      1  mail
f-oil   tty7    1:44pm   35      2      1  -csh
```

(På en eller annen magisk måte (som vil bli forklart nærmere i avsnittet [Omdirigering av utskrift](#) på side 58) antar vi at utskriften fra programmet w i forrige eksempel nå ligger på filen hva.txt.)

Som opsjon til tail oppgir man hvor mange av de siste linjene man vil se, i dette tilfellet 2.

Det er også mulig å la tail telle linjene fra starten av filen, slik at vi får frem hele filen minus noen linjer i starten. På denne måten kan vi fjerne den to-linjers overskriften som ligger først i hva.txt:

```
hunding: dag> tail +3 hva.txt
fosli   tty00   11:34am    8      34     33  emacs
siri    tty01    1:22pm   20      20    20  emacs k7.tex
istvan  tty02    1:16pm   17       7     1  -csh
evy     tty0e    1:09pm   50       7     3  emacs IN324
arthur  tty0     2:14pm    3       1     1  mail
f-oil   tty7    1:44pm   35       2     1  -csh
```

For å fjerne de første linjene, angir vi altså at vi skal ha med fra og med linje 3. Ytterligere opplysninger om programmet tail kan man få ved å skrive man tail.

6.3.4 Programmet sort

Programmet sort er et sorteringsprogram. Hvis intet annet angis, vil programmet sortere en fil i alfabetisk rekkefølge linje for linje.

³ Husk at filer med navn som slutter på .log blir automatisk fjernet etter tre dager; se avsnittet [Automatisk fjerning av filer](#) på side 52.

Vi antar nå at utskriften fra forrige eksempel (oversikten fra `w` minus de to første linjene) ligger på filen `hva3.txt`. Da kan vi få en sortert utskrift av denne filen ved å benytte `sort`-programmet:

```
hunding: dag> sort hva3.txt
arthur   tty0    2:14pm    3      1      1   mail
evy      tty0e   1:09pm   50      7      3   emacs IN324
f-oil    tty7    1:44pm   35      2      1   -csh
fosli    tty00   11:34am    8     34     33   emacs
istvan   tty02   1:16pm   17      7      1   -csh
siri     tty01   1:22pm   20     20     20   emacs k7.tex
```

Ytterligere opplysninger om programmet `sort` kan man få ved å benytte kommandoen `man sort`.

6.3.5 Programmet `grep`

Programmet `grep`⁴ er et søkeprogram. Det leter i én eller flere filer etter en viss tekst, og skriver ut de linjene hvor teksten forekommer. Ved å gi opsjonen `-n` vil `grep` angi filnavnet og linjenummeret.

Anta at vi ønsker å se i hvilke sammenhenger vi bruker en variabel `x1` i et bestemt JAVA-program. Da kan vi skrive

```
hunding: dag> grep -n 'x1' Tallprog.java
Tallprog.java:14: int      x1, x2, x3, sum;
Tallprog.java:19: for (x1 = 1; x1 <= 6; ++x1)
Tallprog.java:20:     sum = x1+x2+x3;
Tallprog.java:25: for (x1 = 3; x1 < 18; x1 += 2) {
Tallprog.java:33:     System.out.println("x1: " + x1);
Tallprog.java:89:     sum += n[x1];
```

De enkle anførselstegnene (' ') rundt søketeksten er ikke alltid nødvendige, men det er lurt å ta dem med til man er sikker i sin sak. Ytterligere opplysninger om `grep` får man ved å skrive `man grep`.

6.3.6 Komprimering av filer

En måte å spare lagringsplass for filer på er å komprimere dem. Til dette benyttes programmet `gzip`:

```
hunding: dag> ls -l gks.*
-rw-r--r-- 1 dag 36341 Jan  5 15:22 gks.java
hunding: dag> gzip gks.java
hunding: dag> ls -l gks.*
-rw-r--r-- 1 dag 8270 Jan  5 15:22 gks.java.gz
```

Filen `gks.java` på 36 341 byte er borte, men den er erstattet filen `gks.java.gz` på bare 8 270 byte. Vi har altså spart 77% av plassen.

Ønsker vi den opprinnelige filen tilbake, er det bare å benytte kommandoen `gunzip`:

⁴ Navnet kommer fra en kommando '`g/(re)/p`' som finnes i programmet `ed`, det aller første redigeringsprogrammet under UNIX. Kommandoen fant frem til linjer som inneholdt et gitt regulært uttrykk (som på engelsk heter «regular expression» forkortet «re») og skrev dem ut.

Dette er et typisk eksempel på at kommandoer i UNIX ofte har kryptiske navn hvis opprinnelse det er vanskelig å gjette.

```
hunding: dag> gunzip gks.java.gz
hunding: dag> ls -l gks.*
-rw-r--r-- 1 dag 36341 Jan 5 15:22 gks.java
```

6.3.6.1 Komprimering av fil-områder

Det kan også være aktuelt å komprimere *alle* filene i et område og alle underområder. Kommandoen `gzip` kan også gjøre dette hvis vi gir opsjonen `-r`:

```
hunding: dag> gzip -r .
```

Kommandoen `gunzip` med opsjonen `-r` dekomprimerer de samme filene tilbake til normal størrelse igjen.

6.4 Omdirigering av I/U

En av de aller sterkeste finessene i UNIX er mulighetene til å omdirigere innlesning og utskrift (ofte forkortet *I/U* eller «I/O» på engelsk).

6.4.1 Omdirigering av utskrift

Det er svært enkelt å omdirigere utskriften fra et program slik at den kommer til en fil i stedet for på skjermen. I avsnittet [Programmet w](#) på side 53 så vi at kommandoen `w` ga en oversikt på skjermen over hvilke brukere som for øyeblikket var logget inn. Hvis vi ønsker denne oversikten på en fil, kan vi utføre kommandoen på følgende måte:

```
hunding: dag> w >hva.txt
```

Da kommer det ingen utskrift på skjermen, men alt sammen blir i stedet lagt på filen `hva.txt`. Innholdet av denne filen kan så skrives ut:

```
hunding: dag> more hva.txt
2:19pm up 16 hrs, 6 users, load average: 1.21, 1.36, 1.07
User      tty      login@  idle  JCPU  PCPU  what
fosli     tty00    11:34am   8     34    33   emacs
siri      tty01    1:22pm    0     20    20   emacs k7.tex
istvan    tty02    1:16pm   17     7     1   -csh
evy       tty0e    1:09pm   50     7     3   emacs IN324
arthur    tty0     2:14pm    3     1     1   mail
f-oil     tty7     1:44pm   35     2     1   -csh
```

Tegnet `>` angir altså at all utskrift til skjermen blir omdirigert til en angitt fil i stedet. En meget stor fordel med dette er at programmet selv ikke vet noe om denne omdirigeringen; det tror hele tiden det skriver til skjermen i lykkelig uvitenhet om at UNIX sender utskriften til en fil. Dette gjør verden enklere for oss som skal skrive programmer.

6.4.2 Omdirigering av innlesning

På samme måte som tegnet `>` angir omdirigering av utskrift, brukes tegnet `<` til å angi omdirigering av innlesning. De data som programmet forventet å lese fra tastaturet, vil da i stedet bli lest fra den angitt filen.

Anta at vi har et JAVA-program som skal testes ut. Ved kjøringen må vi oppgi f.eks. 8 tall:

```
hunding: dag> java Prog
Skriv 8 tall: 12 -2.1 3.142 -8.2 66.02 32.1 -13 -11.9
Svaret er 42.3401
```

Det er en feil i programmet, så det er aktuelt å kjøre det mange ganger. For å slippe å skrive alle tallene ved hver kjøring, er det i stedet mulig å lage en fil Tall.txt som inneholder tallene, og så sørge for at programmet leser tallene fra filen i stedet for fra tastaturet:

```
hunding: dag> java Prog <Tall.txt
Skriv 8 tall:
Svaret er 42.3401
```

Utskriften til skjermen kommer som før, men all innlesningen skjer fra filen.

6.4.2.1 Data inn til filterprogrammer

Programmet more har vi hittil kalt med en parameter som angir hvilken fil den skal skrive ut, men programmet er mer fleksibelt enn som så.

Som svært mange andre UNIX-kommandoer er more skrevet slik at hvis den ikke får oppgitt noen parametre, vil den lese fra terminalen. Det er man sjelden interessert i, men det åpner muligheten for å omdirigere innlesningen. Følgelig vil disse to kommandoene gi samme resultat:

```
hunding: dag> more hva.txt
 2:19pm up 16 hrs, 6 users, load average: 1.21, 1.36, 1.07
User      tty      login@  idle  JCPU  PCPU  what
fosli     tty00    11:34am   8    34    33   emacs
siri      tty01    1:22pm   20    20    20   emacs k7.tex
istvan    tty02    1:16pm   17     7     1   -csh
evy       tty0e    1:09pm   50     7     3   emacs IN324
arthur    tty0     2:14pm    3     1     1   mail
f-oil     tty7     1:44pm   35     2     1   -csh

hunding: dag> more <hva.txt
 2:19pm up 16 hrs, 6 users, load average: 1.21, 1.36, 1.07
User      tty      login@  idle  JCPU  PCPU  what
fosli     tty00    11:34am   8    34    33   emacs
siri      tty01    1:22pm   20    20    20   emacs k7.tex
istvan    tty02    1:16pm   17     7     1   -csh
evy       tty0e    1:09pm   50     7     3   emacs IN324
arthur    tty0     2:14pm    3     1     1   mail
f-oil     tty7     1:44pm   35     2     1   -csh
```

Denne muligheten åpner for sammenkobling, slik det beskrives i avsnittet [Omdirigering gjennom rør](#) på neste side.

6.4.3 Omdirigering av både innlesning og utskrift

Hittil har det vært vist eksempler på omdirigering av enten innlesning eller utskrift, men det er ikke noe i veien for å omdirigere begge to.

Anta at vi ønsker å sortere en fil slik at resultatet havner på en annen fil. Det ordnes enkelt slik:

```
hunding: dag> sort <usortert.txt >sortert.txt
```

Det er imidlertid ikke mulig å omdirigere til og fra samme fil. Ønsker man å sortere en fil slik at resultatet havner på den opprinnelige filen, er det nødvendig med en hjelpefil:

```
hunding: dag> sort <fil> <hjelpefil>
hunding: dag> del fil
hunding: dag> move hjelpefil fil
```

6.4.4 Omdirigering gjennom rør

Takket være mulighetene for omdirigering av innlesning og utskrift har vi nå muligheter for å sende data fra ett program til et annet. Hvis vi for eksempel ønsker en sortert oversikt over alle som er logget inn, kan vi oppnå dette ved å benytte de fire programmene *w*, *tail*, *sort* og *more*. (Vi ønsker å benytte *more* til utskriften i tilfelle det er flere brukere enn det er plass til på skjermen.)

```
hunding: dag> w >temp1
hunding: dag> tail +3 <temp1> >temp2
hunding: dag> sort <temp2> >temp3
hunding: dag> more <temp3>
arthur  tty00      2:14pm      3      1      1  mail
evy      tty0e      1:09pm     50      7      3  emacs IN324
f-oil    tty07      1:44pm     35      2      1  -csh
fosli    tty00     11:34am      8     34     33  emacs
istvan   tty02      1:16pm     17      7      1  -csh
siri     tty01      1:22pm     20     20     20  emacs k7.tex
```

Programmet *w* lager altså de dataene vi trenger, *tail* fjerner overskriften, *sort* sorterer linjene, og *more* viser dem frem én og én skjermfull.

Dette er kraftige mekanismer, men UNIX har enda mer på lager. Når data går direkte fra én fil til en annen, er det mulig å unngå ekstrasfilene ved å koble kommandoene sammen med såkalte **rør** (på engelsk «pipes»). Vi skriver da

```
hunding: dag> w | tail +3 | sort | more
arthur  tty00      2:14pm      3      1      1  mail
evy      tty0e      1:09pm     50      7      3  emacs IN324
f-oil    tty07      1:44pm     35      2      1  -csh
fosli    tty00     11:34am      8     34     33  emacs
istvan   tty02      1:16pm     17      7      1  -csh
siri     tty01      1:22pm     20     20     20  emacs k7.tex
```

Tegnet «|» er et symbol som viser at det går et rør fra ett program til et annet. De data som *w* lager, går altså gjennom røret til *tail*, resultatet derfra går gjennom et annet rør til *sort*, og resultatet fra *sort* går gjennom et tredje rør til *more* som skriver ut svaret.

Det er flere fordeler med rør fremfor filer:

- Det går raskere med rør.
- Brukeren slipper å sitte igjen med mange arbeidsfiler som må fjernes.
- De enkelte programmene forbundet med rør utføres parallelt, slik at det siste programmet kan begynne å gjøre noe før det første er ferdig.

6.4.4.1 Et eksempel

Anta at jeg ønsker å vite hvor mange brukere som har kjørt på den maskinen jeg er logget inn på. Til hjelp har jeg følgende standard UNIX-programmer:

cut kan plukke ut visse kolonner i en linje med tekst; for eksempel vil

```
hunding: dag> cut -f 1 -d ' ' fil
```

skrive ut første kolonne av alle linjene i *fil*. Opsjonen «-d ' » angir at kolonnene er skilt med blanke.

last forteller hvem som sist var logget inn;⁵ parameteren angir hvor mange man vil ha skrevet ut:

```
hunding: dag> last -3
dag      pts/2      Thu Jul 18 09:15  still logged in
doru-cat dtlocal   Thu Jul 18 08:33  still logged in
thomas   dtlocal   Wed Jul 17 21:09 - 07:07 (09:58)
```

Hvis man ikke gir noen parameter, får man hele listen; den er ofte på mange tusen linjer.

sort sorterer som tidligere nevnt linjene i en fil.

uniq sjekker om linjer er like. Hvis det kommer to eller flere like linjer etter hverandre, skrives kun én av dem ut.

wc («word count») teller antall linjer, ord og tegn i en fil og skriver ut svaret; for eksempel forteller

```
hunding: dag> last | wc -l
4570
```

at det er 4570 linjer i last-loggen; altså har det vært 4570 innlogginger på maskinen. (Antallet ord og tegn interesserer oss ikke denne gangen.) Opsjonen -l forteller wc at vi bare er interessert i antallet linjer.

Full dokumentasjon om alle disse kommandoene får man ved å bruke man-kommandoen.

Utstyrt med disse programmene kan vi sette opp en rørledning som gjør jobben for oss:

- 1) last gir en oversikt over alle innlogginger på maskinen.
- 2) cut fjerner alt unntatt første kolonne hvor brukernavnet står.
- 3) sort sorterer brukernavnene.
- 4) uniq fjerner duplikater.
- 5) wc teller hvor mange brukere som har vært logget inn (siden det nå står ett navn per linje).

⁵ Hva er forskjellen på w og last? w forteller hvem som er logget inn *akkurat nå* mens last kjenner til alle som *har vært* logget inn.

Kommandolinjen ser slik ut:

```
hunding: dag> last | cut -f 1 -d ' ' | sort | uniq | wc -l
541
```

Nå vet vi at 541 ulike brukere har vært inne på denne maskinen.

6.5 Kommando-filer

Det er nå vist at det mulig å skrive svært kraftige og spesialiserte kommandoer under UNIX, men de kan fort bli lange. Dessuten er det ikke alltid like lett å huske hva kommandoene heter eller hvilke opsjoner man skal ta med.

Dette avhjelpes ved å benytte **kommandofiler**, som er filer som inneholder kommandosekvenser. Når en slik kommandofil utføres, blir kommandoene i filen utført nøyaktig som om de ble skrevet inn fra tastaturet.

Siden en kommandofil kalles som alle andre kommandoer, nemlig ved å nevne filnavnet, utvider vi mengden av tilgjengelige kommandoer når vi definerer en kommandofil.

6.5.1 Definisjon av kommando-filer

I forrige avsnitt ble det vist et eksempel på en kommandolinje som hentet frem en liste over innloggede brukere, sorterte den, og skrev den ut. Hvis dette er noe vi ønsker å gjøre ofte, kan vi lage en kommandofil som gjør akkurat dette. Filen heter `sortw`; den lages med et vanlig tekstredigeringsprogram (som for eksempel Emacs) og kan se slik ut:

```
hunding: dag> more sortw
#!/store/bin/bash
echo 'Disse er logget inn:'
w | tail +3 | sort | more
```

Denne kommandofilen består av 3 linjer:

- 1) Alle kommandofiler bør starte med en linje som ser nøyaktig slik ut:

```
#!/store/bin/bash
```

Den forteller at kommandofilen skal tolkes av den kommandotolkeren som heter `bash`, og dette er den samme som tolker de kommandoene vi gir i kommandovinduet.
- 2) Her utføres kommandoen `echo`, som skriver ut den teksten som følger etter på linjen. Denne utskriften er selvfølgelig ikke nødvendig, men er tatt med for å gi brukeren litt informasjon.
- 3) Her er selve den kommandolinjen du kjenner igjen fra forrige avsnitt.

Før kommandofilen kan eksekveres, må vi fortelle systemet at den er eksekverbar. Dette gjøres ved å forandre beskyttelsen på filen ved å bruke kommandoen `chmod`; se avsnittet [Endring av fil-beskyttelse](#) på side 46 for en repetisjon av hvorledes den virker.

```
hunding: dag> chmod u+x sortw
```

6.5.2 Utførelse av kommando-fil

En kommandofil utføres like enkelt som andre programmer, nemlig ved å skrive filnavnet:

```
hunding: dag> sortw
Disse er logget inn:
arthur  tty0      2:14pm      3      1      1  mail
evy      tty0e     1:09pm     50      7      3  emacs IN324
f-oil    tty07     1:44pm     35      2      1  -csh
fosli    tty00     11:34am      8     34     33  emacs
istvan   tty02     1:16pm     17      7      1  -csh
siri     tty01     1:22pm     20     20     20  emacs k7.tex
```

6.5.3 Parametre til kommando-filer

Vanlige systemkommandoer kan gis parametre, og det kan kommandofiler også. Notasjonen er som følger:

- \$1 betegner første parameter, \$2 andre parameter og så videre opp til \$9. Forekomster av \$x i kommandofilen vil altså under kjøringen bli erstattet av den tilhørende parameteren; hvis brukeren ikke oppga så mange parametre, vil det bli satt inn en tom tekst.
- \$* betegner listen av alle parametrene.

Som et eksempel kan vi se på en kommando **sortsame** som sorterer en fil. I avsnittet [Programmet sort](#) på side 56 ble det sagt at det ikke var mulig å sortere en fil slik at resultatet havnet på samme fil som den opprinnelige. En kommandofil som gjør dette, er **sortsame**:

```
hunding: dag> more sortsame
#!/local/bin/bash
sort <$1 >temp
del $1
move temp $1
```

Når kommandoen kalles med en parameter:

```
hunding: dag> sortsame Fil1
```

vil **sortsame** sortere **Fil1** og legge resultatet på hjelpefilen **temp**. Etterpå blir **Fil1** fjernet og navnet på **temp** endret til **Fil1**.

6.6 Prosesskontroll

Det vanlige under UNIX (og andre operativsystemer) er at det går nøyaktig én prosess for hver bruker når man kjører fra en terminal og én prosess per vindu når man kjører på en maskin med flere vinduer. Under UNIX behøver det imidlertid ikke være slik hele tiden. Der har brukeren muligheter til å la mange prosesser gå i bakgrunnen, altså utføres samtidig, men uten å lese fra terminalen eller vinduet.

6.6.1 Start av bagrunnsprosesser

Ofte ønsker man å starte en prosess som skal gå i bakgrunnen, det vil si uten kontakt med kommandovinduet. Dette ordnes enkelt ved å skrive en '&' sist på kommandolinjen:

```
hunding: dag> java Tall3 &  
[1] 1461
```

(Nummeret 1461 er prosessens *prosessnummer*; alle prosesser i en UNIX-maskin tilordnes et unikt nummer.)

Proessen blir da startet opp i bakgrunnen og vil gå der til den er ferdig eller brukeren stopper den.

6.6.2 Å drepe prosesser

Brukeren kan drepe den aktive prosessen ved å trykke på  + ;⁶ noen ganger er det nødvendig med to eller flere trykk.

Det er også mulig å drepe andre prosesser en selv har startet, men da trenger man å vite prosessens nummer. Den enkleste måte å få vite dette på, er å bruke kommandoen ps:

```
hunding: dag> ps  
  PID TT      S   TIME COMMAND  
 10155 pts/1   S    0:00 -bash  
 10463 pts/1   S    0:00 sh -c /local/bin/texmf/xdvi unix.dvi  
 10464 pts/1   S    0:19 xdvi.bin -name xdvi -s 6 unix.dvi  
 10880 pts/1   S    0:00 acroread forkurs-2.pdf  
 10430 pts/3   S    0:00 rlogin tyrfing -8 -l dag5
```

Under COMMAND står angitt hvilken kommando som startet prosessen, og under PID står dens nummer. Ønsker jeg for eksempel å drepe prosessen som kjører acroread bruker jeg kommandoen kill:

```
hunding: dag> kill 10880
```

Å drepe prosesser med kill er en *nødløsning* man kun bør ty til når intet annet duger og programmet ikke lar seg stoppe på noen annen måte. Det er fort gjort å ødelegge mer enn man er klar over ved bruk av kill. (Trøsten er at det er umulig å ødelegge for andre brukere.)

6.7 Inn- og utlogging

Både inn- og utlogging har vært omtalt tidligere; her skal det bare tas opp et par ting man bør vite nå når man er blitt litt mer erfaren.

6.7.1 Utidig avslutning av jobber

Når man logger ut fra systemet, det vil si klikker på logout-ikonet, vil man drepe alle vinduene på skjermen, uansett hva de holder på med. For eksempel vil Emacs bli drept uten at filer blir gjemt unna, og andre kommandovinduer kan bli avbrutt midt i utførelsen av noe viktig. Det beste er å avslutte alle vinduer før man logger ut.

⁶ Noen programmer, som for eksempel Emacs, har omdefinert tasten  + ; slike programmer må avsluttes på annen måte.

6.7.2 Bakgrunnsprosesser

Prosesser som går i bakgrunnen skaper ingen problemer når man logger ut; de bare fortsetter å gå selv om brukeren som startet dem er logget ut. På denne måten er det mulig å starte en langvarig jobb før man logger ut, så vil den gå til den er ferdig.

Det eneste som man må tenke på i denne sammenhengen, er at prosessene i bakgrunnen vil bli drept hvis de forsøker å skrive til terminalen; det må være slik for det ville bli kaos hvis skjermen din skulle motta utskrift fra en bruker som forlot terminalen for en halv time siden. Følgelig bør man sørge for at utskriften omdirigeres til en fil. Kommandoen for å starte en kompilering blir da:

```
hunding: dag> java Mittprog > feil &
```

så vil utskriften (deriblant feilmeldingene) bli lagt på filen feil.

6.7.3 Kommando-filer ved inn- og utlogging

Enhver bruker har på sitt hovedområde noen spesielle kommandofiler som utføres ved inn- og utlogging:

.login utføres hver gang brukeren logger inn.

.envir utføres også når man logger inn.

.bashrc utføres hver gang det opprettes et nytt kommandovindu som benytter kommandotolkeren bash. Følgelig utføres den også ved innlogging.

.logout utføres ved hver utlogging.

(Siden disse filene har navn som starter med '.', vil de ikke tas med i en vanlig oversikt over filområdet. Bruk opsjonen `-a` som nevnt i avsnittet [Informasjon om filer](#) på side 39.)

Disse filene er den enkelte brukers eiendom, så det står ham eller henne fritt å endre dem etter eget ønske, men man bør tenke seg svært grundig om for man gjør det. Første grunnregel er:

Ingenting av det som opprinnelig står i disse filene må fjernes eller endres! Uheldig endringer kan gjøre det umulig å logge inn.

Det brukerne relativt sikkert *kan* gjøre, er å legge noen nye linjer der det er satt av plass til det i `.login`, `.envir`, `.bashrc` eller `.logout`. Ønsker man for eksempel et visdomsord med seg på veien, kan man legge en linje med

```
fortune
```

```
eller
```

```
bibel
```

i .logout eller .login.

Flere opplysningener om disse oppsettfilene og hva som er standard oppsett her ved instituttet kan man få ved å skrive `help oppsett`.

Kapittel 7

Når alt går galt

Av og til bare skjærer ting seg, og datamaskiner er intet unntak. Hva gjør man da?

7.1 Hjelp fra andre

Som Ifi-bruker er du heldigvis i et miljø hvor folk er vant med å hjelpe hverandre.

- Alle studenter er plassert i grupper i forbindelse med de kursene de tar, og gruppelærerne er tilgjengelige for spørsmål. Derfor er det naturlig å ta opp problemer med gruppelæreren i første omgang.
- En annen kompetent person å spørre hvis du har problemer, er terminalvakten. På mange terminalstuer er det en slik til stede når terminalstuen er åpen, og det hører med til deres arbeidsoppgaver å svare på spørsmål.
- Det er heller ingen ting i veien for å legge problemet ditt frem for personen ved siden av deg. Det er ikke sikkert vedkommende kan svare, men det koster ikke noe å forsøke.

7.2 Problemer med oppsettet

Når man endrer oppsettet, kan det fort skje at man gjør noe dumt og ting slutter å fungere. Heldigvis kan man enkelt få tilbake standardoppsettet ved å gi kommandoen

```
hunding: dag> installsetup
```

Så får man lese dokumentasjonen en gang til, tenke seg om tre ganger og så prøve igjen.

Hva skjedde med de feilaktige oppsettfilene som man tross alt har lagt ned en del arbeid i? De blir liggende, men nå med «.old» sist i navnet.¹

¹ Husk at disse filene har navn som starter med et punktum, så man må bruke opsjonen -a til ls for å se dem. Dette ble omtalt i avsnittet [Informasjon om filer](#) på side 39.

7.2.1 I verste fall

Hvis oppsettfeilen er alvorlig, kan det hende at man ikke får logget inn i det hele tatt. I så fall må man ty til et lite trikk:

Når man har skrevet passordet i innloggingsvinduet, trykker man *ikke* på , men på  i stedet!

Da får man opp kun et lite kommandovindu og der kan man for eksempel gi kommandoen `installsetup`.

7.3 Når filer forsvinner

Fra tid til annen skjer det at filer blir borte eller ødelegges; dette kan skyldes alt fra brukerfeil til svakheter i systemprogrammer eller strømbrudd. Derfor tas det hver natt kopi av alle brukerfiler.²

Trenger man å få hentet en fil fra sikkerhetskopien, sender man en høflig bønn med data om filen (navn, filområde og dato den sist ble endret) til restore@usit.uio.no som har ansvaret for diskene til laveregradsbrukerne; hovedfagsstudenter og ansatte skriver til drift@ifi.uio.no.

² Det ville ta altfor lang tid å ta kopi av alle brukerfilene, så derfor tar man bare kopi av de som er endret det siste døgnet. Ved å lete bakover i arkivene kan man finne de aktuelle filene.

Oversikt over kommandoer

bash, [21](#)
bibel, [65](#)
bofh, [48](#)

cal, [8](#)
cc, [26](#)
cd, [42](#)
chgrp, [48](#)
chmod, [46](#), [62](#)
copy, [23](#), [42](#)
cp, [23](#)
csh, [21](#)
cut, [61](#)

date, [7](#)
del, [23](#)
du, [30](#)

echo, [62](#)
ed, [57](#)
exit, [13](#), [56](#)

fortune, [9](#), [65](#)
fvwm, [33](#)

gcc, [26](#)
grep, [57](#)
groups, [47](#)
gunzip, [57](#), [58](#)
gzip, [57](#), [58](#)

hjelpe, [10–12](#)

infmap, [28](#)
installsetup, [67](#)

java, [26](#)
javac, [25](#)

kill, [64](#)

last, [61](#)
less, [11](#)
ln, [50](#)
lpq, [25](#)
lprm, [25](#)
ls, [9](#), [22](#), [39](#), [50](#)
lsdel, [23](#)

man, [11](#)

mkdir, [41](#)
more, [10](#), [22](#)
move, [24](#), [42](#)
mozilla, [37](#)
mv, [24](#)

photo, [55](#)
pine, [17](#)
preview, [22](#)
print, [12](#), [24](#)
ps, [64](#)
pwd, [42](#)

rm, [23](#)
rmdir, [42](#)

sh, [21](#)
sort, [56](#), [61](#)

tail, [56](#)

umask, [49](#)
undel, [23](#)
uniq, [61](#)

w, [53](#), [61](#)
wc, [61](#)