

# **Introduktion till LTH:s Linuxdatorer**

**Lunds universitet**

**Version 1.93  
25 maj 2023**



# Innehåll

<b>Förord</b>	<b>v</b>	3.4 Alias . . . . .	11
<b>1 Grunderna</b>	<b>1</b>	3.5 Initieringsfiler . . . . .	12
1.1 Inledning . . . . .	1	3.6 Kommandotolken . . . . .	12
Datorerna . . . . .	1	3.7 Inloggning på andradatorer . . . . .	12
Linuxbaserade operativsystem . . . . .	1	3.8 Processkontroll . . . . .	12
Linux och terminalen . . . . .	1	3.9 Ordbehandling . . . . .	12
Fönstersystemet X . . . . .	2	L <sup>A</sup> T <sub>E</sub> X . . . . .	13
1.2 Inloggning . . . . .	2	LibreOffice . . . . .	13
1.3 Introduktion till Linux . . . . .	2	<b>4 Hemifrån</b>	<b>15</b>
Musoperationer . . . . .	2	4.1 E-post hemifrån . . . . .	15
Val av språk . . . . .	2	4.2 Komma åt filer . . . . .	15
Skärmläsning . . . . .	2	4.3 Köra program hemifrån . . . . .	15
1.4 Passerkort . . . . .	2	4.4 X-server . . . . .	15
1.5 Lösenord . . . . .	2	4.5 Program hemma . . . . .	15
Att byta lösenord . . . . .	2	4.6 Hjälp och kontakter . . . . .	15
1.6 Filer och kataloger . . . . .	3	Nyheter . . . . .	16
Att starta terminalen . . . . .	3	LTH:s studentsidor . . . . .	16
Att skapa filer . . . . .	3	4.7 Kursanmälan . . . . .	16
Att titta i filer . . . . .	3	4.8 Andra tips . . . . .	16
Bläddra i filer . . . . .	3	4.9 Sektionerna . . . . .	16
Att lista filer . . . . .	3	<b>5 Övningsuppgifter</b>	<b>17</b>
Att flytta, döpa om och ta bort filer . . . . .	3	<b>A Lösningar till övningsuppgifterna</b>	<b>19</b>
Kataloger . . . . .	4	<b>Sakregister</b>	<b>21</b>
Att skapa och ta bort kataloger . . . . .	4	<b>Figurer</b>	
Att byta aktuell katalog . . . . .	4	1.1 Katalogträd för en användare . . . . .	4
Att använda kataloger . . . . .	4	1.2 Del av ett katalogträd i ett Linuxsystem . . . . .	5
Absoluta och relativa sökvägar . . . . .	4	<b>Tabeller</b>	
Wildcardcards . . . . .	5	1.1 Förkortningar i filnamn . . . . .	5
1.7 Hjälp . . . . .	5	2.1 Vanliga filnamn . . . . .	7
1.8 Datordriftgruppen . . . . .	5	2.2 Sätt att dirigera om I/O . . . . .	8
Felanmälan . . . . .	5		
<b>2 Påbyggnad</b>	<b>7</b>		
2.1 Skrivare . . . . .	7		
Att skriva ut . . . . .	7		
Skrivarköer . . . . .	7		
Papper . . . . .	7		
2.2 Filnamn . . . . .	7		
Punktfiler . . . . .	7		
2.3 Kommandon . . . . .	7		
Format . . . . .	7		
Starta program . . . . .	8		
Var finns programmet? . . . . .	8		
Filnamnskomplettering . . . . .	8		
2.4 Omdirigering av I/O . . . . .	8		
Pipe . . . . .	9		
2.5 E-post . . . . .	9		
2.6 Information om filer . . . . .	9		
2.7 Säkerhetskopiering . . . . .	9		
2.8 Komprimering av filer . . . . .	9		
2.9 Filskydd . . . . .	10		
Kataloger . . . . .	10		
<b>3 Överkurs</b>	<b>11</b>		
3.1 Java i terminalen . . . . .	11		
3.2 Scala i terminalen . . . . .	11		
3.3 Specialtecken . . . . .	11		



# Förord

Den här handledningen är en kort introduktion till Linuxbase-  
rade operativsystem och hur man kan använda dem.

## Handledningens uppläggning

Denna handledning består av fyra kapitel. Kapitlen *Grunderna* och *Påbyggnad* innehåller ungefär det som man kan förväntas kunna efter första veckan. Kapitlet *Överkurs* behandlar saker som inte är nödvändiga att lära sig den första veckan, men som kan vara bra att kunna för att utnyttja datorerna effektivt under utbildningen. Kapitlet *Hemifrån* innehåller en del tips om hur man kan komma åt skolans datorer utifrån.

Handledningen innehåller rätt mycket information, och har man inte använt Linux tidigare så är det mycket att sätta sig in i. Det är därför lämpligt att fortsätta på egen hand med det som inte hinns med.

## Övningsexempel

På sidan 17 finns det en del övningsuppgifter, som kan vara nyttiga att arbeta sig igenom om man vill träna.

Det finns dock inte övningar på allting som handledningen omfattar, så läsaren uppmanas att själv prova kommandon allt eftersom de introduceras i texten.

För de övningar där det är tillämpligt finns svar i appendix A.

## Simulerad text

När handledningen visar exempel på in- och utmatning i ett terminalfönster ser det ut så här:

```
lo-1{e01nn}: ls
a.out          uppg2          uppg3.pas~
uppg1          uppg2.pas
uppg1.pas     uppg3.pas
lo-1{e01nn}: echo "hej hopp i lingonskogen"
hej hopp i lingonskogen
```

*lo-1* är datorns namn och *e01nn* är den inloggade användaren. Den text som användaren själv skriver in markeras med fetstil (här kommandot `ls` för att läsa en katalog och `echo` för att skriva ut text). Övrig text är sådant som datorn skriver till skärmen.

## Tekniska detaljer

Denna handledning är producerad med typsättningsprogrammet L<sup>A</sup>T<sub>E</sub>X. De teckensnitt som har använts är Times, Avant Garde och Courier. Skärmbilderna är dumpade från en av våra datorer med programmet *import* och sedan redigerade med bildbehandlingsprogrammet GIMP.

## Kommentarer välkomna

Har ni synpunkter på hur handledningen kan bli bättre, så tar vi gärna emot dem, helst via e-post till [datateknik@kansli.lth.se](mailto:datateknik@kansli.lth.se) eller till [infocom@kansli.lth.se](mailto:infocom@kansli.lth.se).



# Kapitel 1

## Grunderna

### 1.1 Inledning

#### Datorerna

Linuxdatorerna som används på LTH är vanliga PC-maskiner av olika modeller och åldrar. Vi har 13 olika Linuxrum med vardera 10–15 arbetsstationer:

**Venus (E:0422)**

**Mars (E:0421)**

**Jupiter (E:0420)**

Rummen finns i norra delen av E-husets källare.

**Hacke (E:0153)**

**Panter (E:0154-E:0155)**

**Lo (E:0210-0211)**

**Val (E:0507-E:0508)**

**Falk (E:0509-e:0510)**

**Varg (E:0518-E:0519)**

Rum i södra delen av E-husets källare.

**Alfa (E:2119)**

**Beta (E:2118)**

**Gamma (E:2121)**

På andra våningen i E-huset.

**Fys:H321**

På Fysicum.

I E-huset finns även ett antal rum med datorer som kör Windows, och i Matematikhuset finns också ett Windowsrum. Windowsdatorerna behandlas inte alls i denna handledning.

LTH har även studentdatorer i Studiecetrum, M-huset, V-huset, A-huset, IKDC, Kemicentrum och Fysicum. Med undantag för sal H321 på Fysicum så kör samtliga av dessa datorer Windows. Det är alltså bara i E-huset och på Fysicum som det finns Linuxdatorer.

Som student på LTH kan man använda datorer i alla husen, men programvarorna i respektive hus är anpassade till undervisningen i huset. De övriga husen saknar alltså en del av de program som ni kan behöva, och på motsvarande sätt finns till exempel inte M-husets CAD-program i E-huset.

Datorerna i LTH:s studiecetrum har inga husspecifika programvaror, utan endast vår grundinstallation. De är i första hand tänkta för enklare uppgifter, som till exempel rapportskrivning.

Sedan finns det några datorrum som tillhör institutioner, men som är kopplade till vårt gemensamma datorsystem för att underlätta för studenterna. Sådana datorer hittar man bland annat på Fysicum, reglerteknik och elektro- och informationsteknik (EIT).

#### Linuxbaserade operativsystem

Ett operativsystem är en uppsättning program som administrerar datorns interna arbete och erbjuder användaren ett antal färdiga kommandon. En central del av ett operativsystem är kärnan (på engelska *kernel*), som hanterar de mest grundläggande uppgifterna såsom exempelvis kommunikationen med hårdvaran och schemaläggning av processer.

Våra datorer använder operativsystemet Ubuntu, som är baserat på operativsystemkärnan Linux. Linuxbaserade operativsystem tillhör vad som brukar kallas de Unix-liknande operativsystemen, då de är baserade på samma konventioner och fungerar på i princip samma sätt som det äldre operativsystemet Unix.

Unix började utvecklas på Bell Laboratories i USA 1969, och är idag ett mycket moget och väl fungerande operativsystem. Unix och Unix-liknande system finns i ett antal olika varianter; Linux är till exempel en helt fri variant av ett Unix-liknande system vars källkod är öppen (det vill säga fritt tillgänglig). Sett ur den vanliga användarens synvinkel är dock de olika Unix-dialekterna mycket lika. Det är oftast bara avancerade användare som märker skillnaderna. **Trots att vi i den här handledningen talar om Linux är den alltså även fullt applicerbar på andra Unix-liknande system.**

Andra mer eller mindre kända Unix-liknande system är MacOS / OSX, diverse BSD-varianter, Solaris, med flera.

#### Linux och terminalen

I denna introduktion så kommer ni att bekanta er med Linux och dess terminal. Terminalen är ett textbaserat program som visar alla sina utskrifter i text och man styr den genom att skriva in textkommandon. Terminalen är ett naturligt redskap för programmerare, då det finns många program som inte har något grafiskt användargränssnitt men som kan köras i terminalen. Detta ger programmeraren en mycket större verktygslåda än en vanlig datoranvändare. Det kan kännas svårt, komplicerat och skrämmande att arbeta med en terminal ifall man inte har använt den förut, men man lär sig efterhand. Denna handledning ger en stabil grund för vidare lärande.

Linux är utvecklat på ett sätt som gör att terminalen är väldigt kraftfull. Man kan göra i stort sett allt via terminalen. Vissa saker kan vara enklare eller bekvämare att göra med ett grafiskt användargränssnitt, men det finns ändå många fördelar med terminalen, till exempel så kan man i ett och samma kommando skicka data mellan olika program om man vill behandla datan på flera sätt. Det går snabbt att göra saker och om man utvecklar ett program så behöver man inte skriva ett användargränssnitt för att testa programmet.

Ni kommer under er utbildning läsa kurser där man skriver avancerade program som inte kommer att ha ett användargränssnitt och det är då bra att ha bekantat sig med hur terminalen fungerar.

## Fönstersystemet X

De program som gör att man kan arbeta med olika fönster på den grafiska bildskärmen kallas för ett *fönstersystem*. Det system som vi använder heter *X Window System, Version 11*, eller kort och gott *X*. Det förekommer även andra förkortningar som *X11* eller *X.org* (efter organisationen som utvecklar systemet). Ibland används även den felaktiga benämningen *X Windows*.

Det program som ger fönstersystemet sitt karaktäristiska utseende kallas för *fönsterhanterare* eller *window manager*. Det ansvarar bland annat för att rita ramar runt fönster och göra menyer, medan varje program ansvarar för resten av utseendet.

I dag går man ofta ett steg längre och använder något som kallas *skrivbordsmiljö* (*desktop environment*). Med en sådan så får alla program ett gemensamt utseende och kan även kommunicera med varandra bättre än om man bara har en fönsterhanterare.

## 1.2 Inloggning

För att kunna använda datorn måste man vara registrerad som användare i datorsystemet. Det brukar kallas för att ha ett *konto*.

En användare identifieras med ett unikt användarnamn (*username*) och ett lösenord (*password*).

Alla studenter har förhoppningsvis redan fått sitt konto skickat via e-post. Annars kan man hämta ut det på Studiecetrum (prata med din labbhandledare). Observera att det inte går att logga in på datorerna förrän man har hämtat ut sitt LU-kort.

## 1.3 Introduktion till Linux

Om du skrev rätt användarnamn och lösenord slutförs inloggningen på några sekunder.

På skärmen finns nu några *ikoner*, och längst till vänster en så kallad *panel*.

## Musoperationer

Musen har tre knappar och fungerar ungefär som i Windows, men med en stor skillnad: För att klippa och klistra behöver man bara markera texten med vänsterknappen, sedan klistrar man in den genom att klicka på mittknappen (rullhjulet). I de flesta program går det även att klippa och klistra med tangentbordsoperationer, precis som i andra operativsystem. Markera en bit text. Använd sedan *Ctrl-C* för att kopiera den eller *Ctrl-X* för att klippa ut den. Flytta sedan muspekaren dit du vill ha texten och tryck *Ctrl-V* för att klistra in texten.

## Val av språk

Eftersom vi har en hel del gäststudenter på LTH så har vi valt att ha engelska som standardspråk på datorerna. Vill man ha ett svenskt användargränssnitt så väljer man det på inloggnings-skärmen.

Om man väljer ett annat språk så ändras det i de flesta programmen. Det finns dock en del program som inte är översatta.

## Skärmläsning

I menyn kan man låsa skärmen manuellt. Den låser sig även efter ett tag om man inte gör någonting. För att låsa upp igen rör man på musen och får upp en ruta där man skriver in sitt lösenord. Det är inte tillåtet att låsa skärmen under längre perioder. I så fall ska man i stället logga ut så att någon annan kan använda datorn. Som tumregel kan man säga att 15 minuter kan vara ok, men det är absolut inte är tillåtet att reservera en dator och sedan gå på en föreläsning i två timmar.

## 1.4 Passerkort

För att kunna komma in i datorrummen behöver man ett passerkort med en personlig kod. Passerkortet får man på studiecentrum. Det fungerar även till ytterdörrarna, så att man kan komma in i i huset på kvällar och helger. Kortet fungerar även som ID-kort, och på kvällar och helger kan väktarna kräva att man visar upp sitt kort.

## 1.5 Lösenord

Ditt användarnamn och lösenord är nyckeln till ditt datorkonto. Det är viktigt att du håller ditt lösenord hemligt så att ingen annan kan komma åt din laborationsrapport eller göra något otillåtet i ditt namn. **Avslöja aldrig ditt lösenord för någon.**

Ibland kan det förekomma mailutskick som säger att man ska skriva in sitt lösenord på en websida för att få behålla sitt mailkonto. Vi skickar **aldrig** ut mail och ber studenterna skriva sina lösenord på en extern websida. Det är bara ett sätt för spammare att komma över konton som sen kan användas till att skicka ut massor av spam! Låt dig alltså inte luras av de här mailen.

## Att byta lösenord

För att inte lösenorden ska bli för lättgissade finns vissa krav på hur ett lösenord ska se ut. De måste vara minst åtta tecken långa, och innehålla tecken ur tre av följande fyra grupper:

- små bokstäver (gemener)
- stora bokstäver (versaler)
- siffror
- specialtecken

Lösenordsbyte görs på <https://passport.lu.se/>

När man har bytt sitt lösenord kan det ta någon minut innan det fungerar överallt.



## 1.6 Filer och kataloger

Hittills har vi bara tittat på grunderna i användargränssnittet. Nu ska vi börja titta närmare på styrkorna med Linux terminal. Ni får nu bekanta er med en uppsättning enkla kommandon som man skriver i ett terminalfönster. Varje kommando motsvarar ett program som finns som standard i Unix-liknande operativsystem. Var för sig är dessa kommandon inte så kraftfulla, men genom att kombinera dem kan man göra riktigt avancerade saker. (Det här går vi in på mer i detalj i nästa kapitel.)

Det är skillnad mellan stora och små bokstäver, både i filnamn och kommandon, så tänk på det när ni testat kommandon.

### Att starta terminalen

Det finns flera olika sätt att starta terminalen. Ett sätt är att högerklicka med musen när muspekaren befinner sig på skärmbakgrunden ("skrivbordsytan"). Då får du upp en pop-up-meny i vilken du kan välja "Open Terminal". Alternativt kan du använda knappkombinationen *Ctrl+Alt+T* (Håll ner knapparna märkta Ctrl och Alt samtidigt som du trycker på T) som brukar fungera på de flesta linuxvarianterna inklusive den vi använder.

När du har lyckats starta en terminal så befinner du dig i din hem-mapp. Med kommandot *ls* kan du läsa vad som finns i mappen. I följande kapitel kommer vi att gå igenom lite grundkommandon för navigering och filhantering. Gå igenom dem noga ifall du inte redan känner dig säker på hur du manövrerar terminalen.

### Att skapa filer

Ett sätt att skriva in text i en fil är att använda kommandot *cat*. Det är en förkortning av *concatenate* som betyder att slå ihop. Kommandot kan användas för att slå ihop flera filer, men i sin enklaste form läser det rader från tangentbordet och skriver dem till terminalen:

```
lo-1{e01nn}: cat
Hej hopp i lingonskogen
Hej hopp i lingonskogen
Detta är den andra raden av tre
Detta är den andra raden av tre
Nu är det dags att avsluta och gå vidare
Nu är det dags att avsluta och gå vidare
<CONTROL-D>
lo-1{e01nn}:
```

För att avsluta inmatningen av text skriver man ett *filslutstecken* (*end-of-file*) först på en rad. I Linux är det tecknet som heter *<CONTROL-D>*, som man får fram genom att hålla nere tangenten märkt CONTROL eller CTRL samtidigt som man trycker på tangenten D. Nu är det ganska meningslöst att skriva text från tangentbordet till skärmen. Vi vill kunna spara texten i en fil. Det gör man genom att använda tecknet *>* för att tala om för Linux att dirigera om utmatningen från ett program till en fil:

```
lo-1{e01nn}: cat > nisse
Nils Nilsson
Gatvägen 12
123 45 Lundfanstorp
<CONTROL-D>
lo-1{e01nn}:
```

Här ser man att terminalen inte upprepar varje mening (som den gjorde i förra exemplet). Detta är för att utskriften från

*cat* dirigeras om (med hjälp av *'>'*) till filen *nisse*, istället för att skrivas ut i fönstret. Svenska bokstäver och bokstäver med accenter ska man helst undvika i filnamn. Mycket av utvecklingen sker i USA, och det är inte alltid utvecklingarna tänker på att programmen ska fungera med våra bokstäver.

### Att titta i filer

För att skriva ut innehållet i en fil på skärmen använder man också programmet *cat*. Genom att skriva ett filnamn efter *cat* läser programmet från filen i stället för från tangentbordet:

```
lo-1{e01nn}: cat nisse
Nils Nilsson
Gatvägen 12
123 45 Lundfanstorp
lo-1{e01nn}:
```

### Bläddra i filer

Nu har vi skrivit ut filer på skärmen med *cat*. Det har nackdelen att man inte kan titta på stora filer. Prova till exempel att skriva ut systemets ordlista med *cat*:

```
lo-1{e01nn}: cat /usr/share/dict/words
```

För att kunna titta på en så stor fil finns programmet *less*. För att kunna titta på systemets ordlista utan att den rullar av skärmen skriver vi:

```
lo-1{e01nn}: less /usr/share/dict/words
```

Nu kan vi gå framåt i texten, sida för sida, med *mellanslag* eller *f*, och bakåt med *b*. Man kan även scrola i texten med hjälp av piltangenterna eller mushjulet. För att avsluta *less* trycker man *q*. *less* har inbyggd hjälp som man får om man trycker *h*. *Less* är inte standard i alla Unix-liknande system, men det finns alltid ett program som heter *more* som fungerar ungefär likadant, men har några färre finesser.

### Att lista filer

Om man inte kommer ihåg vad man kallade filen kan man använda kommandot *ls* vilket är en förkortning för *list*. *ls* skriver ut namnen på alla filer:

```
lo-1{e01nn}: ls
Desktop/ Documents/ Downloads/ Pictures/
Videos/ nisse
lo-1{e01nn}:
```

Filer som slutar med snedstreck, t ex *Documents/*, är kataloger. Katalogerna ovan har systemet skapat åt dig. Vi åter kommer till dessa senare.

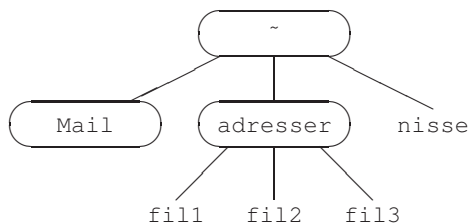
### Att flytta, döpa om och ta bort filer

Nu ska vi titta på tre kommandon som heter *cp*, *mv* och *rm*. Det är förkortningar för *copy*, *move* respektive *remove*.

Med kommandot

```
lo-1{e01nn}: cp nisse nisse2
```

skapar man en kopia av filen *nisse* som heter *nisse2*. Om man i stället skriver



Figur 1.1: Katalogträd för en användare

```
lo-1{e01nn}: mv nisse nisse2
```

flyttar man, alltså döper om, nisse till nisse2. Då finns alltså inte filen nisse kvar. Genom att skriva

```
lo-1{e01nn}: rm nisse2
```

tar man bort filen nisse2.

## Kataloger

Alla filer lagras i *kataloger* (på engelska heter det *directory*). I en katalog kan det finnas underkataloger i flera nivåer. Man brukar kalla det hela för ett katalogträd eftersom det ser ut som ett träd (med roten uppåt).

Varje användare har en egen *hemkatalog*. Det är där man befinner sig när man precis har loggat in. De filer som vi skapade i den förra övningen låg alla i hemkatalogen.

I figur 1.1 finns ett exempel på hur en användares katalogträd kan se ut. Anledningen till att själva hemkatalogen skrivs som tilde (~) återkommer vi till senare.

## Att skapa och ta bort kataloger

För att skapa en katalog använder man kommandot *mkdir* som betyder *make directory*. Med *rmdir* (*remove directory*) tar man bort katalogen.

Om man t ex vill skapa en katalog för att lagra adresser till sina vänner skriver man:

```
lo-1{e01nn}: mkdir adresser
```

Nu kan vi använda *ls* för att se hur det ser ut:

```
lo-1{e01nn}: ls
Mail/      adresser/  nisse
```

Lägg märke till att det finns ett snedstreck sist på den nya katalogen. Det ingår inte i filnamnet, utan det är *ls* som lägger till det för att visa att det är en katalog och inte en vanlig fil. På samma sätt får man en asterisk (\*) sist på ett filnamn om filen är ett exekverbart program (t ex programmet *rm* eller ett program som man har gjort själv i någon kurs). På nyare system är det även vanligt att olika typer av filer/kataloger har olika färg på den text som visas av *ls*.

## Att byta aktuell katalog

Den aktuella katalogen, alltså där man befinner sig just nu, heter på engelska *working directory*. För att se var i filsystemet man befinner sig använder man kommandot *pwd* som betyder *print working directory*.

```
lo-1{e01nn}: pwd
/h/d8/a/e01nn
```

Om man vill byta aktuell katalog använder man kommandot *cd* (*change directory*).

```
lo-1{e01nn}: pwd
/h/d8/a/e01nn
lo-1{e01nn}: cd adresser
lo-1{e01nn}: pwd
/h/d8/a/e01nn/adresser
lo-1{e01nn}: cd ..
lo-1{e01nn}: pwd
/h/d8/a/e01nn
```

Som ni ser använde vi *..* som argument till *cd* för att gå upp en nivå i filträdet. Just i detta fall kunde vi ha skrivit *cd* utan argument. Då kommer man nämligen alltid till hemkatalogen.

## Att använda kataloger

Nu kan vi använda *cp* och *mv* för att kopiera eller flytta filer till den nya katalogen. Om man skriver

```
lo-1{e01nn}: cp nisse adresser
```

så upptäcker *cp* att destinationen är en katalog och skapar en ny fil som heter likadant som originalet, dvs *nisse*, men som ligger i katalogen *adresser*. Nu kan vi titta på den nya filen

```
lo-1{e01nn}: cat adresser/nisse
Nils Nilsson
Gatvågen 12
123 45 Lundfanstorp
```

Som ni ser använder man snedstreck (/) för att skilja katalog och filnamn åt. Därför kan aldrig ett snedstreck ingå i ett filnamn.

Om man vill kalla den kopierade filen något annat än originalet skriver man t ex

```
lo-1{e01nn}: cp nisse adresser/nisse2
```

Det fungerar likadant med *mv*, dvs om destinationen är en katalog får filen samma namn i den nya katalogen.

Man kan även använda *mv* för att döpa om en katalog:

```
lo-1{e01nn}: mv adresser addresses
```

För att se innehållet i en underkatalog använder man *ls* men lägger till katalogens namn:

```
lo-1{e01nn}: ls addresses
nisse      nisse2
```

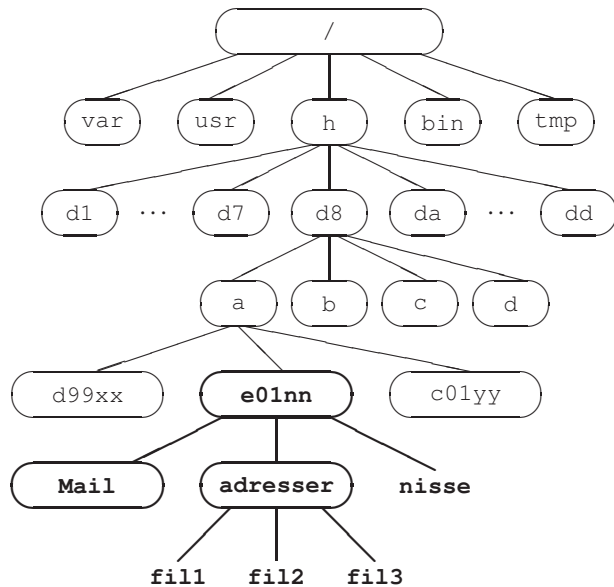
## Absoluta och relativa sökvägar

Katalogträdet i figur 1.1 är egentligen bara en del av systemets stora filträd. I figur 1.2 ser vi det insatt i sitt sammanhang tillsammans med en del andra kataloger i Linuxsystemet.

Längst upp i trädet finns den så kallade *rotkatalogen* eller kort och gott roten. Den betecknas med ett ensamt snedstreck. I just detta speciella fall är alltså / namnet på en katalog i stället för att skilja katalog- och filnamn åt.

När vi skrev *adresser/nisse* var det vad man kallar en *relativ sökväg*, dvs namnet utgår från den aktuella katalogen. Motsatsen är en *absolut sökväg*. Den utgår från roten och samma fil som ovan heter då */h/d8/a/e01nn/adresser/nisse*. Sökväg heter förresten *path* på engelska.

Den absoluta sökvägen till hemkatalogen för *e01nn* är */h/d8/a/e01nn*. Anledningen att vi skrev den som tilde (~) är att man kan använda förkortningar i filnamn (se tabell 1.1).



Figur 1.2: Del av ett katalogträd i ett Linuxsystem

~	min egen hemkatalog.
~user	hemkatalog för användaren med användarnamnet <i>user</i> .
.	Aktuell katalog.
..	Katalogen ovanför aktuell katalog.

Tabell 1.1: Förkortningar i filnamn

Med denna nya kunskap kan vi skriva sökvägen till katalogen *adresser* på ett par olika sätt:

```

adresser
~/adresser
~/e01nn/adresser
/h/d8/a/e01nn/adresser
./adresser
../../../../a/e01nn/adresser

```

## Wildcardcards

Det finns två tecken som är speciella i filnamn, nämligen *asterisk* (\*) och *frågetecken* (?). Dessa har en särskild egenskap, nämligen att de kan motsvara många olika tecken. Därför kallas de för *jokrar* eller *wildcards*. ? motsvarar ett godtyckligt tecken, medan \* motsvarar noll eller flera godtyckliga tecken. Med hjälp av dessa tecken kan man alltså räkna upp flera filer på en kommandorad utan att behöva skriva så mycket.

Om vi har följande filer i en katalog:

```

prog1.p   prog2.p   uppgift.txt
prog1.o   prog2.o   prog
profile.txt  inpris.dat  utpris.dat

```

så kan vi använda kommandona till vänster i stället för de till höger:

```

rm prog?.p   rm prog1.p prog2.p
rm pr*.*     rm profile.txt prog1.o prog1.p prog2.o prog2.p
rm *.o       rm prog1.o prog2.o

```

Observera att Linux antar att man vet vad gör när man skriver ett kommando. Skriver man *rm \** så kommer **alla** filer att försvinna utan att man får en enda fråga. **Var alltså försiktig med wildcardcards.**

## 1.7 Hjälp

I Linuxsystem finns det normalt alltid textbaserade manualer till alla program. Programmet som används heter *man*. Om man t ex vill lära sig allt om *ls* skriver man

```
lo-1{e01nn}: man ls
```

så får man fram den kompletta referensmanualen för *ls*. Manualen navigeras lättast med piltangenterna och när man är klar trycker man *q* (quit) för att avsluta.

Det finns även en del hjälp på [www.student.lth.se/datorsupport/](http://www.student.lth.se/datorsupport/). Här hittar man information om e-post, egna websidor och annat, och på förstasidan finns det driftinformation.

Om ni inte själva lyckas hitta lösningen på ett problem är det dags att prata med någon som kanske vet mer. Har problemet att göra med en uppgift som man ska lösa i en kurs, bör man prata med institutionen som ger kursen. Om man däremot misstänker att det är något fel på själva datorsystemet ska man prata med *Datordriftgruppen (DDG)*.

## 1.8 Datordriftgruppen

Datordriftgruppen LTH eller kort och gott *DDG* har till uppgift att sköta undervisningsdatorerna på LTH och en del annat. Totalt är vi 10 personer.

### Felanmälan

Om man vill rapportera ett fel behöver man inte besöka DDG personligen. Det går också bra att anmäla fel genom att skicka mail till [support@lth.se](mailto:support@lth.se).

Tänk på att berätta utförligt om felet, då ökar chansen att det blir åtgärdat. Ta med eventuella felutskrift från datorn i din felrapport.

Vi har inte möjlighet att gå runt och kontrollera att alla datorer fungerar som de ska, så därför är vi beroende av felrapporter för att kunna åtgärda trasiga datorer.

Tro inte att någon annan redan har rapporterat felet. Om alla resonerar så kan datorerna vara trasiga i flera veckor innan vi upptäcker det. Vi tar hellre emot flera anmälningar av samma fel, än inga alls.



## Kapitel 2

# Påbyggnad

## 2.1 Skrivare

I de flesta datorrum finns en laserskrivare. För att kunna skriva ut så kopplar man sitt kontokort till ett skrivarkonto. Sedan drar universitetet pengar allt eftersom utskriften görs. Detta systemet kallas för PaperCut. Registrering av kontokort gör man på:

<http://www.print.lu.se/papercut/>

Utskriftssystemet är gemensamt för hela Lunds universitet och går alltså även att använda utanför LTH.

### Att skriva ut

Från programmen kan man oftast skriva ut via menyerna, men för att skriva ut en fil från kommandoraden använder man kommandot *lp* som står för line print. För att skriva ut en fil *nisse* skriver man

```
lo-1{e01nn}: lp nisse
```

då kommer utskriften att skickas till PaperCut och köas upp i väntan på att du aktiverar utskriften på valfri skrivare, se nedan.

För att få lite snyggare formatering av utskrifterna kan man använda programmet *a2ps*. Till exempel gör man så här för att skriva ut ett dokument med två sidor på varje papper:

```
lo-1{e01nn}: a2ps -2 nisse
```

För att se vad man kan göra med *a2ps* i övrigt, skriv

```
lo-1{e01nn}: a2ps --help
```

Självklart går det bra att skriva ut från andra program som stödjer utskrift så som Adobe Reader eller en texteditor. För att skriva ut från dessa program så trycker man på *skriv ut* i programmen och sedan väljer PaperCut som skrivare.

### Skrivarköer

När man använder *lp* skickas inte utskriften direkt till skrivaren. I stället köas den upp i utskriftssystemet (PaperCut). För att verkligen få ut utskriften så går du till önskad skrivare och följer instruktionerna som är anslagna där för att aktivera utskriften.

### Papper

DDG fyller på skrivarna med jämna mellanrum, men ibland kan det bli tomt i pappersmagasinen. I så fall är det bara att säga till.

<code>fil.java</code>	Javaprogram
<code>fil.c</code>	C-program
<code>fil.C</code>	C++-program
<code>fil.cc</code>	C++-program
<code>fil.txt</code>	textfil
<code>fil.ps</code>	grafikformatet postscript
<code>fil.pdf</code>	grafikformatet PDF
<code>fil.tex</code>	källkod till typsättningsprogrammet $\LaTeX$

Tabell 2.1: Vanliga filnamn

## 2.2 Filnamn

I Linux kan ett filnamn se ut nästan hur som helst. Det finns ingen begränsning på hur långa namnen får vara, och formatet kan se ut nästan hur som helst. I vissa andra operativsystem måste filnamn se ut som *namn.typ*, men i Linux behöver man inte skriva så. Det är dock vanligt att man använder den konstruktionen för att tala om vad det är för slags fil. Några exempel hittar ni i tabell 2.1. För att undvika problem bör man begränsa sig till att använda följande tecken i filnamn: *a-z*, *A-Z*, *0-9*, *.*, *-*, *\_*. Det är i och för sig tillåtet att använda t ex svenska bokstäver i filnamn, men eftersom många program är skrivna av amerikaner fungerar det inte alltid.

### Punktfiler

Om en fil har en punkt först i namnet brukar den kallas för en *punktfil*. Sådana filer syns inte när man skriver *ls*. Dessa filer innehåller inställningar till program och annat som mest är i vägen när man listar sina filer. Det går dock att se punktfilerna om man skriver *ls -a*.

## 2.3 Kommandon

Linux skiljer inte på *kommandon* och *program*. Alla de kommandon som vi har använt hittills (*cat*, *ls*, *mv*, *cp*, *rm*, ...) är fristående program, och inte på något sätt inbyggda i systemet. Att köra programmet *cat* fungerar alltså likadant som att köra ett program som man har skrivit själv.

### Format

De flesta kommandon består av kommandots eller programets namn, följt av eventuella *optioner* och *argument*. Om vi till exempel skriver

```
lo-1{e01nn}: ls -l adresser
```

så är:

- ls** namnet på programmet som ska köras.
- l** en option som i detta fall talar om att ls ska använda ett längre format när den visar innehållet.
- adresser** ett argument, i det här fallet namnet på den katalog som vi vill se innehållet i.

Vissa program kan använda flera optioner samtidigt. Då kan man oftast skriva ihop dem. Ordningen mellan optionerna är inte heller viktig för det mesta. Följande rader är helt ekvivalenta:

```
ls -l -F -C -i adresser
ls -lF -C -i adresser
ls -lFCi adresser
ls -ilCF adresser
```

Tänk på att det är skillnad mellan små och stora bokstäver även i optioner. `ls -a` är inte samma sak som `ls -A`.

## Starta program

För att starta ett program skriver man helt enkelt programmets namn. Det har vi gjort många gånger nu, med program som `rm` eller `cat`.

När vi skapade ett nytt terminalfönster körde vi också program, men i stället för att skriva programmets namn bad vi fönsterhanteraren att köra programmet åt oss genom att välja en rad i en meny. Man kan själv starta nya fönster genom att skriva programmets namn i ett terminalfönster.

När man startar program i terminalen så kan det hända att terminalfönstret läser sig tills programmet har avslutats. Detta löser man genom att starta programmet i *bakgrunden* och detta gör man genom att skriva tecknet `&` efter kommandot, t ex

```
lo-1{e01nn}: gedit &
```

för att starta texteditorn `gedit`.

## Var finns programmet?

Programmet `cat` består av en fil som heter `cat`. Men det finns ingen sådan fil i din hemkatalog. Hur bär sig då Linux åt för att hitta alla program?

I en variabel som heter *PATH* lagras sökvägen till ett antal kataloger där systemet letar efter program. Du kan se vilka dessa kataloger är genom att skriva ut innehållet i variabeln:

```
lo-1{e01nn}: echo $PATH
```

Som ni ser så finns det rätt många kataloger med program. Bland dem brukar det finnas några som ligger i hemkatalogen. Där kan man lägga program som man vill kunna köra utan att skriva fullständig sökväg.

Observera att den katalog man befinner sig i *inte* ingår i sökvägen. Vill man köra ett program som ligger i den aktuella katalogen måste man skriva *.programnamn*.

## Filnamnskomplettering

Att använda långa filnamn kan vara bra eftersom man kan beskriva innehållet i en fil på ett entydigt sätt. Det är däremot jobbigt att skriva in det långa namnet när man ska använda filen.

Om *p* är ett program och *fil* är en fil.

<code>p &gt; fil</code>	Koppla stdout till <i>fil</i> . Filen skapas om den inte redan finns. Om filen redan finns skrivs den över.
<code>p &gt;&gt; fil</code>	Som ovan, men en existerande fil skrivs inte över. I stället läggs den nya utmatningen till i slutet av filen.
<code>p 2&gt; fil</code>	Koppla stderr till <i>fil</i> .
<code>p &gt;&amp; fil</code>	Koppla både stdout och stderr till <i>fil</i> .
<code>p &lt; fil</code>	Koppla stdin till <i>fil</i> . Gör att ett program läser sin inmatning från <i>fil</i> .
<code>p   prog</code>	koppla stdout till stdin för programmet <i>prog</i> .
<code>p  &amp; prog</code>	koppla stdout och stderr till stdin för programmet <i>prog</i> .

Tabell 2.2: Sätt att dirigera om I/O

Då kan man använda sig av filnamnskomplettering. Om man skriver början av filnamnet och sedan trycker på tabulatortangenten (märkt *Tab* eller `→`) fylls resten av filnamnet i automatiskt. Om det man skrev inte var tillräckligt för att unikt peka ut en fil, alltså om flera filers namn börjar likadant, så kompletteras så långt som det går. Då kan man antingen skriva några tecken till och trycka på *Tab* igen, eller trycka på *Tab* två gånger för att få en lista med möjliga alternativ.

Så här kan det se ut (*Tab*-tryckningar indikeras med `<TAB>`):

```
lo-1{e01nn}: ls
Example.java      ProgramTester.java
Main.java         ProgramUtilities.java
ProgramModel.java
lo-1{e01nn}: less P <TAB>
lo-1{e01nn}: less Program
lo-1{e01nn}: less Program <TAB> <TAB>
ProgramModel.java ProgramUtilities.java
ProgramTester.java
lo-1{e01nn}: less ProgramU <TAB>
lo-1{e01nn}: less ProgramUtilities.java
```

## 2.4 Omdirigering av I/O

Först några ord om rubriken. I/O betyder Input/Output, dvs in- och utmatning till eller från ett program.

Alla program har tillgång till tre kanaler för I/O: en inkanal och två utkanaler. Inkanalen kallas *stdin* (standard input) och är normalt kopplad till tangentbordet. Utkanalerna heter *stdout* (standard output) och *stderr* (standard error) och är normalt kopplade till terminalfönstret. På *stderr* kommer felutskriften och på *stdout* kommer normal utmatning från ett program.

När man skriver ett kommando kan man med skrivsätten i tabell 2.2 ange att dessa tre kanaler ska kopplas någon annanstans än till tangentbord eller skärm.

Den första formen av omdirigering har vi redan stött på när vi skrev till filer med `cat`, men resten kräver nog en del förklaring. För att göra detta ska vi introducera ett nytt program som heter `wc` (word count). Det räknar tecken, ord och rader i en fil:

```
lo-1{e01nn}: wc nisse
3      7      45 nisse
```

Utmatningen berättar att filen *nisse* innehåller 3 rader, 7 ord och 45 tecken. Nu ska vi dirigera om utmatningen från `wc` till en fil:

```
lo-1{e01nn}: wc nisse > utfil
lo-1{e01nn}: cat utfil
3      7      45 nisse
```



I detta fallet hamnade utmatningen från programmet `wc` i filen `utfil`. Men det är bara `stdout` vi har dirigerat om, inte `stderr`, vilket vi kan se om vi försöker räkna ord i en fil som inte finns:

```
lo-1{e01nn}: wc finns-inte > utfil
wc: cannot open finns-inte
lo-1{e01nn}: cat utfil
lo-1{e01nn}:
```

Som ni ser fick vi en tom `utfil` eftersom `wc` inte skrev något på `stdout`. Däremot skrevs det ett felmeddelande på `stderr`.

`wc` läser normalt sin inmatning från tangentbordet (tills vi avslutar med `<CONTROL-D>`). Till exempel:

```
lo-1{e01nn}: wc
Detta är den första raden
Rad nummer två.

Till slut rad 4, efter en tom rad 3.
<CONTROL-D>
4      17      83
```

Hittills har vi dock gett `wc` ett filnamn som argument för att den ska läsa från filen i stället. Det är en funktion som finns inbyggt i `wc`, men alla program har inte den möjligheten. Istället kan man dirigera om inmatningen så att den kommer från en fil, istället för från tangentbordet, för att uppnå samma sak:

```
lo-1{e01nn}: wc < nisse
3      7      45
```

Nu skrev inte `wc` ut något filnamn eftersom programmet inte vet vad filen heter, det fick bara *innehållet* i filen till `stdin`.

## Pipe

De båda sista formerna av omdirigering i tabell 2.2 är lite speciella. Tecknet `|` brukar kallas för *pipe* eftersom det skapar en slags rörledning för data.

När programmet `ls` skriver till skärmen får man filnamnen uppställda i kolumner. Om man dirigerar om utmatningen skriver `ls` i stället bara ett filnamn per rad. Det kan man använda för att räkna filer.

Om vi till exempel vill räkna antalet filer i en katalog kan vi använda en temporär fil, och skriva

```
lo-1{e01nn}: ls > tmp
lo-1{e01nn}: wc < tmp
7      7      48
lo-1{e01nn}: rm tmp
```

Det blir rätt mycket att skriva för en sådan enkel uppgift. Dessutom måste vi skapa en fil och vänta med att räkna orden tills `ls` är klar med att lista dem. Sen blir det en fil för mycket i sammanställningen eftersom den temporära filen också räknas.

I stället kan man med en *pipe* koppla ihop två kommandon. Tecknet `|` anger att `stdout` från det första programmet ska kopplas ihop med `stdin` för det andra:

```
lo-1{e01nn}: ls | wc
6      6      44
```

Detta är mycket enklare att skriva och har en finess som inte är helt uppenbar, nämligen att båda programmen körs samtidigt. Om man kopplar samman två program som tar lång tid kan det gå betydligt fortare att använda en *pipe* än en temporär fil.

Man kan givetvis koppla ihop fler än två program och dessutom kombinera det med att läsa från filer:

```
lo-1{e01nn}: cat < nisse | wc | cat > utfil
lo-1{e01nn}: cat utfil
3      7      45
```

Program som `cat` och `wc` läser sin inmatning från `stdin` och skriver sin utmatning till `stdout`. Program av denna typen lämpar sig väl för sammankoppling med en *pipe*, och brukar kallas för *filter*.

## 2.5 E-post

När du börjar på LTH får du ett eget e-postkonto. Du bör ha fått separat information om hur du gör för att logga in och läsa din e-post.

E-postadressen är studentens officiella adress. Lärare och administratörer på LTH använder den adressen för att komma i kontakt med studenterna. Därför är du **skyldig att bevaka din e-post hos oss**, antingen genom att läsa den i webmailen eller genom att vidarebefordra den till en annan adress. Fråga din övningsledare om du vill ha hjälp att forwarda din e-post till just din adress.

## 2.6 Information om filer

Genom att ge programmet `ls` optionen `-l` listar den filerna i ett annat format:

```
lo-1{e01nn}: ls -l nisse
-rw----- 1 e01nn students 45 Jul 28 16:58 nisse
```

Den första kolumnen är information om filskyddet. Det återkommer vi till på sidan 10. Sedan följer en etta som vi bortser från (skriv *man ls* om du är intresserad). Därefter följer filens ägare, gruppen som filen tillhör, filens storlek, datum då den senast ändrades och filens namn.

## 2.7 Säkerhetskopiering

Med jämna mellanrum görs en säkerhetskopiering (*backup* på engelska) av hela datorsystemet. Då skrivs alla ändrade filer till en bandspelare eller till en server med mycket disk. Kopieringen görs i första hand för att kunna återställa systemet om ett skivminne skulle gå sönder. I mån av tid hjälper DDG till med att läsa tillbaka filer som en användare "råkat radera" eller som "bara försvann". Det är dock ingenting som ni kan räkna med, så om ni har viktiga filer som ni absolut inte vill bli av med kan det vara klokt att göra en kopia på ett USB-minne.

För att be att få tillbaka en borttappad fil, skicka ett brev till `support@lth.se` och tala om vad filen heter, när den försvann och när den senast ändrades.

## 2.8 Komprimering av filer

Varje användare har ett begränsat utrymme tillgängligt för att lagra filer. Hur mycket det är, och hur mycket man har använt kan man ta reda på med ett program som heter *quota*:

```
lo-1{e01nn}: quota
Disk quotas for e01nn (uid 25002):
Filesystem usage quota limit timeleft files quota limit
/a/d8      2096 25000 50000      137 10000 20000
```

Först finns siffrorna för det utrymme som filerna tar upp (räknat i kilobyte, dvs 1024 tecken), därefter kommer en grupp siffror som gäller antalet filer. Skillnaden mellan *quota* och *limit* ligger i att man tillfälligt får gå upp till den övre gränsen. För vårt system ligger tidsgränsen på 12 timmar.

Om man har gått över den tillåtna gränsen så kan man inte längre logga in på normalt sätt. Genom att markera *Starta endast enkelt fönster* nere i vänstra hörnet på inloggningsskärmen kan man logga in tillfälligt för att städa.

Tänk på att program som du har kompillerat tar stor plats. Där emot tar inte källkoden (programtexten) upp mycket utrymme. Ta bort alla exekverbara program när programmeringsuppgifterna är färdiga så sparar du mycket utrymme. Detta gäller för de flesta programmeringsspråken, dock inte Java.

Om man har mycket ont om plats och inte kan städa bort något kan man komprimera en del av sina filer. Det gör man med ett program som heter *gzip*. Det brukar minst kunna halvera storleken på filerna. För att komprimera filen *nisse* skriver man

```
lo-1{e01nn}: gzip nisse
```

Filen ersätts av den komprimerade filen som i detta fallet heter *nisse.gz*. Den kan man inte använda som vanligt utan att först dekomprimera den. För det ändamålet använder man programmet *gunzip*:

```
lo-1{e01nn}: gunzip nisse
```

Om man vill komprimera eller dekomprimera alla filer i en underkatalog (och dess underkataloger) lägger man till optionen *-r* (recurse) till *gzip* respektive *gunzip*:

```
lo-1{e01nn}: gzip -r adresser
```

En komprimerad textfil kan man titta på utan att packa upp den. Det gör man med *zless* eller *zmore*. Båda programmen gör exakt samma sak, nämligen packar upp filen tillfälligt och visar den i less.

```
lo-1{e01nn}: zmore nisse
```

Observera att den diskkvot som en användare har ska räcka till alla filer som produceras under studietiden. Bara i undantagsfall kan man få mer utrymme att tillgå och i så fall är det bara tillfälligt medan man läser kurser som kräver mycket plats.

Ett bra sätt att ta reda på hur mycket plats ens filer tar är att använda programmet *du* (disk usage):

```
lo-1{e01nn}: du
4      ./Mail/inbox
2      ./Mail/drafts
14     ./Mail
5      ./adresser
2      ./temp
1      ./Calendar
1      ./News
22     ./mozilla
645   .
```

För varje katalog talar *du* om hur många kilobyte filerna tar, och sist av allt kommer summan för alla kataloger.

## 2.9 Filskydd

Nu ska vi återkomma till utmatningen från *ls -l*:

```
lo-1{e01nn}: ls -l nisse
-rw----- 1 e01nn students 45 Jul 28 16:58 nisse
```

Den första kolumnen består av 10 tecken. Det första anger filens typ (de vanligaste är *-* för vanliga filer och *d* för kataloger). De övriga nio tecknen anger hur filskyddet ser ut.

I Linux finns det nämligen möjlighet att skydda filer så att obehöriga inte kan komma åt dem. För varje fil kan man specificera huruvida filen får läsas, skrivas eller exekveras av användare i olika kategorier. Dessa kategorier är:

- u (user)** Filens ägare
- g (group)** Medlemmar i samma grupp som ägaren. Alla studenter är med i samma grupp (*students*).
- o (others)** Övriga användare

De olika rättigheterna anges med följande bokstäver

- r (read)** Tillstånd att läsa
- w (write)** Tillstånd att skriva
- x (execute)** Tillstånd att exekvera (köra) ett program.

Filskyddet anges i ordningen: användare, grupp och övriga. T ex betyder *-rw-r-----* att ägaren får läsa och skriva filen, att gruppen bara får läsa och att övriga inte har tillgång till filen. (Strecket betyder att motsvarande position är avstängd och finns bara för att det ska vara lättare att läsa filskyddet.)

För att ändra filskyddet använder man kommandot *chmod* som tar två argument:

```
chmod kategorier=rättigheter filnamn
```

Kategorier och rättigheter anges med bokstäverna som förklaras ovan. Rättigheter kan vara tomt om man vill stänga av alla rättigheter för en kategori. Om man t ex vill hindra andra än ägaren att komma åt en fil skriver man:

```
lo-1{e01nn}: chmod go= nisse
```

## Kataloger

För kataloger betyder filskyddet inte riktigt samma sak som för vanliga filer:

- r** Tillstånd att läsa i katalogen, dvs se vilka filer som finns i katalogen (t ex med *ls*).
- w** Tillstånd att skriva i katalogen, dvs skapa filer och ta bort filer. Det är katalogens rättigheter och inte filens som avgör om en användare kan ta bort en fil.
- x** Tillstånd att ha med katalogen i en sökväg. Om en katalog bara har *x* för en viss kategori kan användarna i den kategorin fortfarande komma åt filer i katalogen om de vet vad filerna heter.



## Kapitel 3

# Överkurs

### 3.1 Java i terminalen

Om man vill köra ett program som är i skrivet i Java så behöver man först kompilera det. För att kompilera filen `Main.java` så skriver man:

```
lo-1{e01nn}: javac Main.java
```

och man kommer då få en fil som heter `Main.class`. För att köra programmet skriver man:

```
lo-1{e01nn}: java Main
```

Om ens program består av mer än en källfil så behöver man kompilera dessa också. Om man till exempel har ett program som består av filerna `Main.java` och `Network.java` så kompilerar man dessa tillsammans genom att skriva:

```
lo-1{e01nn}: javac Main.java Network.java
```

Har man många java-filer som man vill inkludera så kan det vara lättast att använda sig av ett wildcard. Det skulle kunna se ut på följande sätt:

```
lo-1{e01nn}: javac *.java
```

Då kommer alla java-filer i mappen att kompileras.

### 3.2 Scala i terminalen

Program skrivna i Scala kan kompileras och köras på motsvarande sätt som om de var skrivna i Java. För att kompilera `Main.scala` så skriver man:

```
lo-1{e01nn}: scalac Main.scala
```

För att köra programmet skriver man sedan (om klassen som innehåller main-metoden heter `Main`):

```
lo-1{e01nn}: scala Main
```

Se kursboken i EDAA45 Programmering, grundkurs för mer information om hur du programmerar i Scala.

### 3.3 Specialtecken

På sidan 3 berättade vi lite om kontrolltecken (som man får fram om man håller nere tangenten `CONTROL` och trycker på en annan tangent). När man skriver kommandon i ett terminalfönster har dessa tecken (och en del andra) en speciell betydelse.

Om man t ex skriver fel så kan man använda nedanstående kommandon för att rätta sina fel innan man trycker *Return*:

**Del** Det tecken man senast skrivit raderas. Med upprepede *Del* kan man radera flera tecken.

**Control-u** Hela raden raderas.

**Control-w** Ordet *före* markören raderas.

**Control-k** Allt *efter* markören raderas.

Man kan använda piltangenterna  $\rightarrow$  och  $\leftarrow$  för att flytta markören på raden. Om man vill köra gamla kommandon igen kan man använda  $\uparrow$  för att få tillbaka dem på kommandoraden. För varje gång man trycker  $\uparrow$  går man ett steg tillbaka i listan med gamla kommandon. Om man går för långt kan man trycka  $\downarrow$  för att gå åt andra hållet.

Man kan även söka efter gamla kommandon om man trycker *Control-r* och sedan de tecken man söker efter. När man har hittat rätt kommando trycker man *Return* för att köra det.

Följande kontrolltecken är också bra att känna till:

**Control-c** Avbryter exekveringen av pågående program. Det kan användas när man ångrar sig efter att ha tryckt *Return*, eller om ett program har hängt sig.

**Control-z** Stoppar exekveringen av ett kommando tillfälligt. Genom att skriva *fg* (foreground) startar man det igen. Man kan även skriva *bg* (background). Då fortsätter programmet att köra i bakgrunden, precis som om man hade startat det med *&* sist på raden.

**Control-d** Filslutstecken. Anger att data är slut vid inmatning från tangentbordet.

**Control-l** Tömmer hela terminalfönstret på text.

Vissa tangenter har extra symboler som t ex `{ } [ ] ~`. Dessa får man fram genom att hålla nere tangenten märkt *Alt Gr* samtidigt som man trycker på respektive tangent. För vissa tecken, t ex `~` (tilde), behöver man också göra ett extra mellanslag innan tecknet visas, eftersom man också kan skriva en bokstav efter för att producera t ex tecknen `ã ñ õ`.

### 3.4 Alias

Genom att definiera *alias* kan man skapa egna förkortningar av långa kommandon. Alias definieras genom att skriva

```
alias namn='sträng'
```

Det får inte finnas några blanktecken före eller efter likhets-tecknet. Om man skriver *alias* utan argument får man en lista över alla alias som finns definierade.

Om man t ex vill kunna skriva *dir* i stället för *ls -l* kan man definiera ett alias:

```
lo-1{e01nn}: alias dir='ls -l'
```

Ett alias finns bara kvar tills man loggar ut, men genom att lägga in dem i initieringsfilen `.bashrc` kan man få dem att definieras varje gång man loggar in (se nedan för mer info).

## 3.5 Initieringsfiler

Många program läser en *initieringsfil* när de startas. Det finns initieringsfiler av två slag: dels filer som innehåller kommandon som exekveras när programmet startas, dels filer som innehåller parametrar som läses av programmet. De vanligaste initieringsfilerna är:

- .bash\_login** Exekveras vid inloggning.
- .bash\_logout** Exekveras vid utloggning.
- .bashrc** Exekveras när man startar ett nytt terminalfönster.

Ett varningens ord är på sin plats: Om man har egna initieringsfiler kan man få problem. I det läget får man tyvärr räkna med att klara sig själv eftersom DDG inte har resurser till att sätta sig in i alla personliga konfigurationer. Kopiera alltså inte kompisens fräcka initieringsfiler om du inte känner att du har kunskaper nog att ta hand om de problem som kan dyka upp.

## 3.6 Kommandotolken

Det program som läser och utför kommandon i ett terminalfönster kallas för *kommandotolk* eller *shell*. Den variant som vi använder heter *bash*.

Alla initieringsfiler som började med `.bash` i tabellen ovan läses av *bash* vid inloggning och/eller vid start av en ny kommandotolk. I de gemensamma initieringsfilerna finns information om vilken typ av saker som är lämpliga att lägga i vilka filer.

Det vanligaste är att man vill ha en egen `.bashrc` för att spara några egna alias och inställningar till *bash*. Här är ett exempel på inställningar som kan vara bra att ha i sin `.bashrc` om man är nybörjare:

```
alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'
setopt noclobber
setopt ignoreeof
```

De tre första raderna gör att *rm*, *cp* och *mv* frågar innan de tar bort eller skriver över en fil. *setopt noclobber* gör att man inte kan skriva över en fil genom att dirigera om I/O. *setopt ignoreeof* gör att man inte kan avsluta ett terminalfönster med *Control-d* utan måste skriva *exit* eller *logout*.

## 3.7 Inloggning på andra datorer

Ibland kan man behöva logga in på en annan dator än den man sitter vid. Det kan vara fallet om man sitter hemma, eller i ett lab på någon annan institution. För detta ändamål använder man programmet *ssh* (secure shell client) genom att skriva:

```
ssh datornamn
```

Då startar man en terminal på den dator som man namngett, och kan köra kommandon och program där precis som vid den egna datorn. Tänk på att man absolut inte får köra tunga program på andra datorer på detta sättet. Då kan man störa den som sitter vid den andra datorn. I värsta fall kan det medföra att en laboration inte går att genomföra. Därför har vi ett antal maskiner som endast är avsedda för inloggningar utifrån.

Om man ansluter sig från en dator som befinner sig utanför vårt nätverk så måste man dessutom använda loginmaskinerna. Datorerna i salarna är nämligen inte tillgängliga utanför LUs nätverk. Man når loginmaskinerna under ett gemensamt namn, `login.student.lth.se`. Hemifrån måste man alltså skriva:

```
ssh username@login.student.lth.se
```

För mer information om inloggning hemifrån, se nästa kapitel.

Man kan ge ett program lite lägre prioritet genom att skriva *nice* först på kommandoraden. Det är att rekommendera om man kör program på någon annan dator än den som man sitter vid. För att t ex räkna ord i en fil utan att störa andra på datorn så mycket skriver man

```
lo-1{e01nn}: nice wc nisse
3          7          45 nisse
```

Om datorn känns ovanligt långsam och man misstänker att någon annan också kör på den kan man använda kommandot *who* för att kontrollera vem som är inloggad på datorn. Tänk dock på att den databas som *who* använder kan bli korrupt ibland, så risken finns att man kan skälla på fel person.

## 3.8 Processkontroll

En dator med Linux kör flera program samtidigt genom att låta varje program exekvera några millisekunder, och sedan byta till nästa program, eller process som det också kallas.

För att se vilka processer man har startat kan man använda programmet *ps* (process status):

```
lo-1{e01nn}: ps
PID TTY      TIME CMD
10083 pts/2    0:00 ps
10073 pts/2    0:01 xcalendar
10061 pts/2    0:01 bash
```

Den första kolumnen visar processens nummer, och den sista det kommando man skrev.

Om man vill ta bort en process som har hängt sig använder man kommandot *kill*, med processens nummer som argument. Om vi t ex vill ta bort den process som kör kalendern skriver vi:

```
lo-1{e01nn}: kill 10073
```

Om processen fortfarande inte vill dö kan man ta i lite mer:

```
lo-1{e01nn}: kill -9 10073
```

För mer information, titta på manualerna till *ps* och *kill*.

## 3.9 Ordbehandling

Man brukar skilja på editering och *ordbehandling*. Editering handlar om att redigera rena textfiler, medan ordbehandling betyder att man skriver formaterad text, ofta med olika tecken-

## L<sup>A</sup>T<sub>E</sub>X

L<sup>A</sup>T<sub>E</sub>X fungerar så att man skriver sin text med formateringskommandon i en vanlig editor. Sedan formaterar man den med kommandot *latex*. Nu kan man studera utseendet på skärmen med *xdvi* eller skriva ut den på en laserskrivare med *dvips*. L<sup>A</sup>T<sub>E</sub>X är mer eller mindre standard som ordbehandlare på tekniska universitet i hela världen. (Den här handledningen är skriven med L<sup>A</sup>T<sub>E</sub>X.)

Vi har en lokal handledning som heter *Att skriva rapporter med L<sup>A</sup>T<sub>E</sub>X*. Den används som kurslitteratur i ettan för C och D, men finns även att köpa hos DDG för 10 kronor.

L<sup>A</sup>T<sub>E</sub>X är fri programvara, och finns för nästan alla datortyper, så den kan man köra även på sin hemdator.

## LibreOffice

För er som trivs bättre med ett system som liknar *Microsoft Word* så finns *LibreOffice* tillgängligt i menyerna. LibreOffice är helt gratis och klarar i allmänhet av att läsa Microsofts dokument. Det kan med andra ord även vara ett alternativ för hemdatorn. LibreOffice finns även för Windows och MacOS.



# Kapitel 4

## Hemifrån

Detta kapitel innehåller lite tips om hur man kan utnyttja datorsystemet hemifrån, samt lite allmänt matnyttig information. Texten bygger till delar på ett tidigare häfte gjort av Erik Månsson och Emma Nilsson-Nyman.

### 4.1 E-post hemifrån

Eftersom Lunds universitets studentmail är webbaserad, så är det enkelt att sköta sin e-post från vilken dator som helst.

### 4.2 Komma åt filer

För att titta på, ladda hem eller ladda upp filer så använder man *scp* (secure copy) eller *sftp* (secure file transfer protocol). Kör man något Unix-liknande operativsystem (Linux, BSD, MacOS / OSX) hemma har man redan dessa program.

Om man vill ladda upp en fil som heter *Main.java* så kan man göra följande:

```
scp ./Main.java username@login.student.lth.se:
```

För de som kör windows rekommenderar vi ett program som heter *winscp*. Det är gratis och kan laddas ned från <http://winscp.net/>. Man kan även använda ett annat program som heter Filezilla (<http://filezilla-project.org/>).

Det går även att komma åt filer med FTPS. Då kör man i stället mot en server som heter *ftp.student.lth.se*. (I vissa programvaror kan det här även kallas FTPES, FTP Secure eller FTP-SSL.)

### 4.3 Köra program hemifrån

Det är givetvis bäst att installera de program man behöver på sin hemdator, men ibland går det inte. Det kan bero på att programmet behöver filer som bara finns på skolan, eller att man av licensskäl inte får köra det hemma.

För att ansluta till datorerna kör man *ssh* (secure shell) mot någon av loginmaskinerna. Tänk på att loginmaskinerna inte är gjorda för tunga och långa programkörningar, så använd dem bara till lättare program.

Har man windows hemma kan man ladda hem ett ssh-program som heter *putty* från [www.chiark.greenend.org.uk/~sgtatham/putty/](http://www.chiark.greenend.org.uk/~sgtatham/putty/).

Starta programmet. Gå in under *Session* i vänsterspalten och fyll i namnet på loginmaskinen under *Host Name (or IP address)*. Kontrollera att *SSH* är markerat som *Protocol*. Klicka

sedan på *Open*. Svara *Yes* i frågerutan och logga sedan in med ditt namn och lösenord.

Om man kör Linux eller MacOS så räcker det med att man skriver `ssh username@login.student.lth.se` i en terminal för att ansluta.

När du nu har anslutit till skolan kan du starta icke X-baserade program (program utan grafik) som till exempel egna scala- eller javaprogram utan grafik. Om du däremot vill starta program med grafik (till exempel en firefox, egna grafiska java-program eller grafer i Matlab) så behöver du en X-server. (Om du försöker starta ett program som kräver X-server så får du felmeddelandet *Error: Can't open display.*)

### 4.4 X-server

Att installera och konfigurera en X-server kräver en del kunskaper. En bra gratis X-server för Windows är *Cygwin/X* som man kan ladda ned på <http://x.cygwin.com/>.

Starta X-servern. Därefter startar du *putty*, går in under fliken *Tunnels* och klickar för *Enable X11 forwarding*. Lämna rutan *location* tom och anslut sedan på vanligt sätt till en logindator.

Ett annat gratisalternativ är *Xming* som finns på <http://www.straightrunning.com/XmingNotes/>.

För MacOS finns X-servern *XQuartz* att ladda ner från <http://www.xquartz.org/>. Den är gratis att installera och använda. När du startar *XQuartz* får du även upp ett terminalfönster. Från detta kan du ansluta till LTHs inloggningsserver med *ssh* på samma sätt som om du kör Linux, se nedan.

Kör du någon variant Linux hemma är det mycket enklare - då har du nämligen redan en X-server. I så fall är det bara att logga in som vanligt med *ssh* men lägga till `-Y` som argument, alltså `ssh -Y username@login.student.lth.se`.

### 4.5 Program hemma

Det finns en del program som kan laddas hem gratis från skolans servrar. Dels från [program.ddg.lth.se](http://program.ddg.lth.se), dels från Studentportalen ([www.student.lu.se](http://www.student.lu.se)).

### 4.6 Hjälp och kontakter

Snabbaste hjälpen får du troligen genom att prata med andra studenter. Du kan även fråga dina lärare eller handledare som säkert kan guida dig i rätt riktning.

## Nyheter

Nyheter om datorerna hittar man på DDG:s driftinfo på <http://www.ddg.lth.se/>. När det är nätverksproblem eller annat som ligger utanför LTH kan man även titta på <http://www.ldc.lu.se/drift>. (LDC är en organisation inom universitetet som bland annat sköter nätverken.)

## LTH:s studentsidor

På <http://www.student.lth.se/> kommer du åt en massa information, bland annat schema, kursplaner och programhemsidor.

## 4.7 Kursanmälan

Inför varje läsperiod måste du anmäla dig till de kurser du vill läsa. Detta gäller dock inte första terminen, men under senare delen av hösten ska du anmäla dig till kurser på vårterminen. Kursanmälan görs i kursanmälningsystemet som finns i Studentportalen (<http://www.student.lu.se/>).

## 4.8 Andra tips

- Kopiera filer till en annan användare genom att skriva `scp fil username@localhost:` och ange den andra användarens lösenord. Glöm inte kolon på slutet. Filen hamnar i användarens hemkatalog.
- Kolla kvarvarande diskutrymme med kommandot *quota*. För att hitta vilka filer eller kataloger som tar upp plats, kör du `-s .??* * | sort -n`
- Om du startar ett grafiskt program utan att skriva `&` sist på kommandoraden kan du trycka Ctrl-Z för att tillfälligt stoppa programmet och sedan `bg` för att fortsätta köra det i bakgrunden. Kommandot `fg` plockar fram programmet och kör det i förgrunden igen (som om man inte hade skrivit `&`).

## 4.9 Sektionerna

Här är länkar till de olika sektionerna:

F/Pi/Nano	<a href="https://www.fsektionen.se/">https://www.fsektionen.se/</a>
E	<a href="https://eee.esek.se/">https://eee.esek.se/</a>
M	<a href="https://www.maskinsektionen.com/">https://www.maskinsektionen.com/</a>
V	<a href="http://www.v-sek.se/">http://www.v-sek.se/</a>
A	<a href="http://www.asektionen.se/">http://www.asektionen.se/</a>
K	<a href="http://www.ksek.se/">http://www.ksek.se/</a>
D/C	<a href="http://www.dsek.se/">http://www.dsek.se/</a>
Dokt	<a href="http://www.dokt.tlth.se/">http://www.dokt.tlth.se/</a>
Ing	<a href="http://www.ingsekt.se/">http://www.ingsekt.se/</a>
W	<a href="http://www.wsek.se/">http://www.wsek.se/</a>
I	<a href="http://www.isek.se/">http://www.isek.se/</a>

## Kapitel 5

# Övningsuppgifter

### Övning 5.1

Logga in på datorn med dina kontouppgifter. Om ditt konto inte fungerar kan din övningsledare hjälpa dig lösa problemet.

### Övning 5.2

Öppna ett terminalfönster genom att trycka på `Ctrl+Alt+T` eller genom att trycka på ikonen längst upp i panelen, sök på *Terminal* och välj terminal.

Öppna sedan en editor (*ikonen längst upp i panelen->Text Editor*). Prova att skriva lite text och kopiera den inom ett fönster och mellan fönstren.

### Övning 5.3

Prova att skriva text med `cat`. Lär dig hur man skriver `<CONTROL-D>` för att avsluta inmatningen.

### Övning 5.4

Skapa en fil som med samma namn som ditt förnamn och mata in ditt namn och adress. Använd gärna `ao` i stället för `åö` om du har svenska bokstäver i ditt förnamn, och undvik accenter.

### Övning 5.5

Skriv ut din fil på skärmen och kontrollera att din adress stämmer.

### Övning 5.6

Hur gör man för att söka efter en sträng i `less`? (Använd den inbyggda hjälpen.)

### Övning 5.7

Prova att byta namn på din fil och skapa ett par kopior av den. Ta sedan bort alla kopior så att det bara finns en fil kvar. Under tiden kan du använda `ls` för att se vilka filer som finns och `cat` för att se deras innehåll.

### Övning 5.8

Skapa en ny katalog som heter `temp`. Skapa en kopia av din adressfil i den nya katalogen. Försök sedan att ta bort katalogen med `rmdir`. Vad händer?

### Övning 5.9

Skapa en underkatalog som heter `slask`. Prova att kopiera och flytta din fil till den nya katalogen, dels under samma namn, dels med nytt namn. Döp sedan om katalogen till `adresser` och ta bort några filer. Använd under tiden `ls` för att se innehållet i katalogen.

### Övning 5.10

Vilken katalog pekar `~/../../../adresser/../../../../` på? Kan ovanstående sökväg förkortas och i så fall hur och varför?

### Övning 5.11

Eftersom en ensam punkt pekar på aktuell katalog kan man inte ha den till mycket, eller...? Försök komma på ett tillfälle när den behövs.

### Övning 5.12

Vilka filer kommer att tas bort om man skriver `rm *pr*` om vi har filerna från kapitlet 1.6 Wildcards?

### Övning 5.13

Ta fram manualen för `rm`. Vad betyder det om man skriver `rm -i` i stället för att bara skriva `rm`?

### Övning 5.14

Använd `ls -a ~` för att se dina filer i din hemkatalog. Hur skiljer sig `ls -a` mot `ls`?

### Övning 5.15

Kör programmet `xcalc` genom att skriva `xcalc` i ditt terminalfönster och trycka *Return*. Vad händer?

### Övning 5.16

Avsluta kalkylatorn och starta den sedan igen genom att skriva `xcalc &`. Kontrollera att du nu kan skriva nya kommandon i terminalfönstret.

### Övning 5.17

Skriv `echo $PATH` för att ta reda på vilka kataloger som finns i din sökväg för program. `echo` är ett program som helt enkelt skriver ut sina argument, prova till exempel med `echo hej` eller `echo *`

### Övning 5.18

Räkna antalet filer i din hemkatalog och i katalogen `adresser`.

### Övning 5.19

Vad betyder utmatningen från följande kommando:  
`wc fil | wc`?





## Bilaga A

# Lösningar till övningsuppgifterna

- 5.4 Använd `cat > namn.txt` för att skapa och skriva till filen
- 5.5 Använd `cat filnamn` för att läsa från filen
- 5.6 Man skriver `/sträng` för att söka efter *sträng*.
- 5.7 använd `mv` för att döpa om (flytta) din fil, och `cp` för att kopiera den
- 5.8 `rmdir` svarar: `rmdir: temp: Directory not empty.`  
 Det går alltså inte att ta bort en katalog som innehåller filer. För att kunna ta bort katalogen måste man skriva `rm temp/fil` och sedan köra `rmdir`.  
 Ett annat sätt är att skriva `rm -r temp`. Det betyder: ta bort katalogen `temp` och *alla* filer och underkataloger som finns i `temp` (`-r` betyder rekursiv). Detta är lite farligt om man råkar skriva fel.
- 5.10 Det är samma sak som din hemkatalog. Eftersom en ensam punkt betyder att man går till den katalog där man redan befinner sig kan alla `./` tas bort. `adresser/..` betyder att man går ner i `adresser` och sedan upp igen. Alltå kan även detta tas bort. Hela sökvägen kan förkortas till `~`.
- 5.11 Om man vill kopiera eller flytta en fil från en annan katalog *till* aktuell katalog utan att döpa om den skriver man t ex:
- ```
cp adresser/nisse .
```
- 5.12 `inpris.dat profile.txt prog prog1.o prog1.p prog2.o prog2.p utpris.dat`
- 5.13 Det får `rm` att fråga om tillstånd innan den tar bort några filer.
- 5.14 `ls -a` visar alla filer som finns i mappen, `ls` visar inte filer som börjar med `.` Punktfiler är ofta konfigurationsfiler för olika program och behöver därför inte alltid visas. Man ser också mapparna `.` och `..` som pekar på den nuvarande mappen respektive mappen som den nuvarande mappen ligger i.
- 5.15 Det dyker upp en kalkylator. `xcalc` är namnet på programmet för en kalkylator.
- 5.18 Du kan använda `ls | grep wc` för att räkna filer i aktuell katalog
- 5.19 Först räknar man antalet ord i *fil*. Därefter räknar man antalet ord i utmatningen från `wc`.



# Sakregister

|                         |          |                          |                  |
|-------------------------|----------|--------------------------|------------------|
| <b>Symboler</b>         |          |                          |                  |
| * .....                 | 4, 5     | echo .....               | 17               |
| .....                   | 5        | email .....              | 9                |
| .....                   | 5        | end-of-file .....        | 3                |
| .....                   | 5        | exit .....               | 12               |
| .bashrc .....           | 12       |                          |                  |
| / .....                 | 4        | <b>F</b>                 |                  |
| < .....                 | 8        | f .....                  | 3                |
| > .....                 | 3, 8     | fg .....                 | 11               |
| >& .....                | 8        | filer                    |                  |
| » .....                 | 8        | filskydd .....           | 10               |
| ? .....                 | 5        | information .....        | 9                |
| ~ .....                 | 5        | komprimera .....         | 10               |
| 2> .....                | 8        | filkatalog .....         | 4                |
|                         |          | filnamn .....            | 7                |
| <b>A</b>                |          | komplettering .....      | 8                |
| a2ps .....              | 7        | filskydd .....           | 10               |
| absolut sökväg .....    | 4        | filslutstecken .....     | 3                |
| aktuell katalog .....   | 4        | filter .....             | 9                |
| alias .....             | 11       | frågetecken .....        | 5                |
| användarnamn .....      | 2        | fönsterhanterare .....   | 2                |
| argument .....          | 7        | fönstersystem .....      | 2                |
| asterisk .....          | 5        |                          |                  |
|                         |          | <b>G</b>                 |                  |
| <b>B</b>                |          | gunzip .....             | 10               |
| b .....                 | 3        | gzip .....               | 10               |
| backup .....            | 9        |                          |                  |
| bakgunden .....         | 8        | <b>H</b>                 |                  |
| bash .....              | 12       | h .....                  | 3                |
| .bash_login .....       | 12       | hemkatalog .....         | 4                |
| .bash_logout .....      | 12       | hjälp .....              | 5                |
| .bashrc .....           | 12       |                          |                  |
| bg .....                | 11       | <b>I</b>                 |                  |
|                         |          | I/O .....                | 8                |
| <b>C</b>                |          | ignoreeof .....          | 12               |
| cat .....               | 3        | ikoner .....             | 2                |
| cd .....                | 4        | initieringsfil .....     | 12               |
| chmod .....             | 10       |                          |                  |
| copy .....              | 3        | <b>J</b>                 |                  |
| cp .....                | 3, 4, 12 | Java i terminalen .....  | 11               |
| Ctrl+Alt+T .....        | 17       | jokrar .....             | 5                |
|                         |          |                          |                  |
| <b>D</b>                |          | <b>K</b>                 |                  |
| datordriftgruppen ..... | 5        | katalog .....            | 4                |
| datorerna .....         | 1        | aktuell .....            | 4                |
| DDG .....               | 5        | filskydd .....           | 10               |
| directory .....         | 4        | skapa .....              | 4                |
| du .....                | 10       | ta bort .....            | 4                |
|                         |          | katalogträd .....        | 4                |
| <b>E</b>                |          | kill .....               | 12               |
| e-post .....            | 9        | kommandon .....          | 7                |
|                         |          |                          |                  |
|                         |          | kommandotolk .....       | 12               |
|                         |          | komprimera filer .....   | 10               |
|                         |          | konto .....              | 2                |
|                         |          |                          |                  |
|                         |          | <b>L</b>                 |                  |
|                         |          | laserskrivare .....      | 7                |
|                         |          | latex .....              | 13               |
|                         |          | less .....               | 3                |
|                         |          | LibreOffice .....        | 13               |
|                         |          | logout .....             | 12               |
|                         |          | lp .....                 | 7                |
|                         |          | ls .....                 | 3, 4, 9          |
|                         |          | lösenord .....           | 2                |
|                         |          |                          |                  |
|                         |          | <b>M</b>                 |                  |
|                         |          | man .....                | 5                |
|                         |          | mellanslag .....         | 3                |
|                         |          | mkdir .....              | 4                |
|                         |          | move .....               | 3                |
|                         |          | musoperationer .....     | 2                |
|                         |          | mv .....                 | 3, 4, 12         |
|                         |          |                          |                  |
|                         |          | <b>N</b>                 |                  |
|                         |          | nice .....               | 12               |
|                         |          | noclobber .....          | 12               |
|                         |          |                          |                  |
|                         |          | <b>O</b>                 |                  |
|                         |          | operativsystem .....     | 1                |
|                         |          | optioner .....           | 7                |
|                         |          | ordbehandling .....      | 12               |
|                         |          |                          |                  |
|                         |          | <b>P</b>                 |                  |
|                         |          | panel .....              | 2                |
|                         |          | papper .....             | 7                |
|                         |          | passerkort .....         | 2                |
|                         |          | password .....           | 2                |
|                         |          | path .....               | <i>se</i> sökväg |
|                         |          | pipe .....               | 9                |
|                         |          | processer .....          | 12               |
|                         |          | program .....            | 7                |
|                         |          | programstart .....       | 7                |
|                         |          | ps .....                 | 12               |
|                         |          | punktfil .....           | 7                |
|                         |          | pwd .....                | 4                |
|                         |          |                          |                  |
|                         |          | <b>Q</b>                 |                  |
|                         |          | q .....                  | 3                |
|                         |          | quota .....              | 9                |
|                         |          |                          |                  |
|                         |          | <b>R</b>                 |                  |
|                         |          | relativ sökväg .....     | 4                |
|                         |          |                          |                  |
|                         |          | remove .....             | 3                |
|                         |          | rm .....                 | 3, 12            |
|                         |          | rmdir .....              | 4                |
|                         |          | rotkatalogen .....       | 4                |
|                         |          |                          |                  |
|                         |          | <b>S</b>                 |                  |
|                         |          | Scala i terminalen ..... | 11               |
|                         |          | setopt .....             | 12               |
|                         |          | shell .....              | 12               |
|                         |          | skriv ut .....           | 7                |
|                         |          | skrivare .....           | 7                |
|                         |          | skrivarkö .....          | 7                |
|                         |          | skrivbordsmiljö .....    | 2                |
|                         |          | Skärmläsning .....       | 2                |
|                         |          | snedstreck .....         | 4                |
|                         |          | specialtecken .....      | 11               |
|                         |          | ssh .....                | 12               |
|                         |          | stderr .....             | 8                |
|                         |          | stdin .....              | 8                |
|                         |          | stdout .....             | 8                |
|                         |          | svenska bokstäver .....  | 3                |
|                         |          | säkerhetskopiering ..... | 9                |
|                         |          | sökväg .....             | 8                |
|                         |          | absolut .....            | 4                |
|                         |          | relativ .....            | 4                |
|                         |          |                          |                  |
|                         |          | <b>T</b>                 |                  |
|                         |          | Terminal .....           | 17               |
|                         |          | tilde .....              | 4                |
|                         |          |                          |                  |
|                         |          | <b>U</b>                 |                  |
|                         |          | underkataloger .....     | 4                |
|                         |          | Unix .....               | 1                |
|                         |          | username .....           | 2                |
|                         |          | utskrift .....           | 7                |
|                         |          |                          |                  |
|                         |          | <b>W</b>                 |                  |
|                         |          | wc .....                 | 8                |
|                         |          | who .....                | 12               |
|                         |          | wildcards .....          | 5                |
|                         |          |                          |                  |
|                         |          | <b>X</b>                 |                  |
|                         |          | X .....                  | 2                |
|                         |          | X Window System .....    | 2                |
|                         |          | X.org .....              | 2                |
|                         |          | X11 .....                | 2                |
|                         |          |                          |                  |
|                         |          | <b>Z</b>                 |                  |
|                         |          | zless .....              | 10               |
|                         |          | zmore .....              | 10               |





