



KTH Datavetenskap
och kommunikation

Lathund i UNIX-användning på Nada och KTH CSC



Stockholm 2010



KTH | Skolan för datavetenskap och kommunikation
SU | Numerisk analys och datalogi

Utgåva 5.0, 2010-08-20

Författare: Per Sedholm

Baserad på tidigare material av Mårten Svantesson, David Vuorio, Lena Nylén, Richard Spångberg, Jörgen Björkner, Anna Herting, Mattias Frånberg

Teckningen på framsidan är gjord av Jesper Ooppelstrup.

Synpunkter och kommentarer mottages tacksamt på system@csc.kth.se

Innehåll

Innehåll	i
Inledning	1
Historik	1
... och så sköter man sig!	2
Läsanvisning	2
1 Inloggning	5
1.1 Fyllagring	5
1.2 Datorsupport	6
2 Grafiskt gränssnitt (GUI)	7
2.1 Mus och tangentbord	7
2.1.1 Mus – alla tre knappar används	7
2.1.2 Tangentbord	7
2.2 Teckenkodning (<i>UTF-8</i> eller <i>Latin-1</i>)	10
3 Ubuntu	12
3.1 Inloggning	12
3.1.1 Felsäker inloggning (vid problem)	12
3.1.2 Om du inte kan logga ut	12
3.2 Gnome Desktop	13
3.2.1 Starta applikationer (program)	13
3.3 Internet och kommunikation	15
3.4 Ordbehandling och kalkyl	15
3.5 Grafik	15
3.6 Texteditor	16
3.7 Flyttbara media (USB-minnen etc)	17
3.7.1 Utskrifter	17
3.8 \LaTeX	18
3.9 Stavningskontroll: spell, stava	19
4 Mac OS X	20
4.1 Inloggning	20
4.2 Finder	20
4.2.1 Flyttbara media (USB-minnen etc)	21
4.2.2 Keychain	21
4.3 Internet	21
4.4 Ordbehandling	22
4.4.1 iWork	22
4.4.2 MS Office 2008	22
4.4.3 Ren text; \LaTeX	23
4.5 Grafik	23
4.5.1 Adobe-sviten: CS5	23
4.5.2 Andra program (urval)	24
4.5.3 Utskrifter	24
5 Solaris	25
5.1 Inloggning	25
5.1.1 Felsäker inloggning (vid problem)	25
5.1.2 Tunna klienter (SunRay-terminaler)	25
5.1.3 Java Desktop System (JDS)	26
5.1.4 Flyttbara media (USB-minnen etc)	27
6 Kommandoraden i UNIX	28
6.1 Grundläggande: skal, prompt, sökväg	29
6.2 Byta lösenord (<code>kpasswd</code>)	29
6.3 Kommandon	30
6.3.1 Förflyttning i filträdet (<code>cd</code> och <code>pwd</code>)	30

6.3.2	Lista filer (<code>ls</code>)	31
6.3.3	Skapa och ta bort kataloger (<code>mkdir</code> och <code>rmdir</code>)	32
6.3.4	Flytta och kopiera filer (<code>mv</code> och <code>cp</code>)	32
6.3.5	Ta bort filer (<code>rm</code>)	32
6.3.6	Visa textfiler (<code>cat</code> , <code>more</code> , <code>less</code>)	33
6.3.7	Utskrifter, utskriftskvoter	33
6.3.8	Symbolisk länk (<code>ln -s</code>)	34
6.3.9	Filrättigheter (<code>fs</code> , <code>chmod</code>)	35
6.3.10	Skriv text (<code>echo</code>)	37
6.3.11	Program och processer (<code>kill</code> , <code>fg</code> , <code>bg</code>)	37
6.4	Begrepp	38
6.4.1	Kommando, argument, väljare	38
6.4.2	Filtrådet	39
7	Fördjupning	42
7.1	Mer om kommandotolken	42
7.1.1	Kommandohistoria	42
7.1.2	<i>Wildcards</i> (jokertecken)	42
7.1.3	Script	43
7.1.4	<i>Tab completion</i> (filnamnsexpansion)	43
7.1.5	Omdirigering	43
7.1.6	Alias och funktioner	45
7.1.7	Exportera miljövariabler	45
7.1.8	Moduler	46
7.2	AFS och Kerberos	46
7.2.1	Biljetter	47
7.2.2	Kvot	47
7.3	Rättigheter	47
7.3.1	Rättighetsgrupper	48
7.3.2	Säkerhetskopior – OldFiles	48
7.3.3	Delade kataloger	49
7.3.4	Olika rättigheter i samma katalog	50
7.4	Hemsida (<code>public_html</code>)	50
7.5	Fjärrinloggning (<code>ssh</code>)	51
7.5.1	Terminalinloggning (SSH)	51
7.5.2	Filöverföring (SFTP, SCP)	51
7.5.3	X11 forwarding	52
7.5.4	Tyngre och långvariga beräkningar	53
7.5.5	Fjärrinloggning med Kerberos	53
8	Emacs	55
8.1	Textredigering, klipp-och-klistra	56
8.2	Söka text	57
8.3	Hantera flera filer	57
8.4	Lägen	57
8.5	Inställningar	58
8.6	Hjälp	58
8.7	Alternativ	58
9	Programmering	59
9.1	Java	59
9.1.1	Integrerad utvecklingsmiljö, IDE (<i>Eclipse</i>)	60
9.2	Python	60
9.2.1	IDE (<i>Idle</i>)	60
9.3	MATLAB	60
9.3.1	Octave	61
9.4	Subversion (<code>svn</code>)	62
10	Dokumentation och hjälp	63
10.1	Man-sidor (<code>man</code> , <code>info</code>)	63
10.2	Online-hjälp	64

10.3	Mänsklig hjälp: handledning	64
10.3.1	Kurshandledning	64
10.3.2	Sima Manager	65
10.3.3	Allmänhandledning	65
10.3.4	Systemgruppen	65
10.3.5	Annan datorsupport	65
10.4	Problemlösning – FAQ	65
10.5	För dig som har ett gammalt konto	67
10.5.1	Konstigt användarnamn	67
10.5.2	"Fel" skal (<i>tcs</i> h eller <i>bash</i>)	67
10.5.3	Gamla inställningar	67
10.5.4	Återställa default-inställningar	67
11	Kursadministration	69
11.1	Mina Sidor	69
11.2	rapp	69
11.3	course	69
11.4	res	69
A	Några termer	72
B	Vanliga tecken	73
C	Program	75
	Sakregister	78
D	Lathund till UNIX-kommandon	81
E	Lathund för Emacs	82

Inledning

Detta kompendium syftar till att ge grundläggande kunskaper om den datormiljö du kommer att använda på KTH CSC, skolan för datavetenskap och kommunikation. KTH CSC använder många olika typer av operativsystem.¹ I datorsalar finns:

- Apple-datorer med Mac OS X
- PC med Microsoft Windows
- PC med Ubuntu Linux
- Sun arbetsstationer med Solaris

Förutom den rent fysiska skillnaden mellan datorerna, så innebär de olika OSen att man oftast inte kan flytta ett program från en datormiljö till en annan. Däremot finns de flesta "standardprogram", såsom exempelvis MATLAB, tillgängliga i alla miljöer.

De flesta har idag någon datorvana, men denna är oftast med MS Windows och/eller Mac OS X, snarare än de UNIX och Linuxsystem som finns tillgängliga i datorsalarna. Tyngdpunkten i detta kompendium är därför att visa hur UNIX och Linux skiljer sig från dessa, snarare än att lära ut allting från grunden. En stor del av kompendiet behandlar det som är gemensamt för UNIX, och UNIX-liknande operativsystem.

Historik

UNIX är en familj av operativsystem vars rötter kan spåras till *Unics*, ett operativsystem som användes på AT&Ts Bell Labs år 1969. För att officiellt klassas som ett UNIXsystem, och använda varumärket UNIX, så måste ett OS i praktiken ha utvecklats från (eller dela de utmärkande dragen hos) antingen V7 (Version 7, Berkeley UNIX, även kallat *BSD-grenen) eller System V (Solaris, HP-UX, etc).

I praktiken så menar de flesta "UNIX-liknande" när man idag pratar om "ett UNIX-system". Alltså, att systemet uppträder som och tillhandahåller de verktyg man kan förvänta sig. Detta kallas ibland för *NIX, för att undvika problem med varumärket UNIX. Linux är ett sådant OS (även om det inte formellt klassificerats som UNIX), liksom de olika BSD-systemen och Mac OS X.

Förutom själva operativsystemets kärna (eng. *kernel*), så behöver man också en uppsättning verktyg (*toolchain*) som låter en göra saker som att skapa och ta bort filer och kataloger, visa ett grafiskt användargränssnitt (GUI, *Graphical User Interface*), kompilera andra program, och så vidare. Till detta hör grafiska program såsom filhanterare². Ett OS består vanligen av en kärna, en grundläggande uppsättning verktyg och andra applikationer (ordbehandlare, webbläsare, etc). En linuxdistribution är baserad på Linuxkärnan och (vanligtvis) GNUs verktyg.

Ubuntu är en linuxdistribution, skapad av det Sydafrikanskt ägda företaget Canonical Ltd. Det är, till stor del, baserat på den tidigare distributionen Debian. Namnet Ubuntu kommer från Zulu-språket.

- [UNIX](http://en.wikipedia.org/wiki/UNIX) (<http://en.wikipedia.org/wiki/UNIX>)

¹Ett operativsystem, eller "OS", är det program som startar när man slår på datorn, och som startar andra program. Det hanterar datorns hårdvara, och sköter också kommunikationen mellan programmen, och över nätverket med andra system.

²MS Windows *Explorer*, Mac OS Xs *Finder*

- [Linux](http://en.wikipedia.org/wiki/Linux) (<http://en.wikipedia.org/wiki/Linux>)
- [GNU-projektet](http://en.wikipedia.org/wiki/GNU_project) (http://en.wikipedia.org/wiki/GNU_project)

... och så sköter man sig!

Alla datoranvändare på CSC KTH har skrivit på högskolans regler, som bland annat säger

Datorresurserna ägs och drivs av KTH för användning i av KTH auktoriserad verksamhet. All annan verksamhet, såsom egen utveckling, är enbart tillåten när den:

- ordinarie användningen inte störs,
- inte innebär brott mot dessa föreskrifter,
- inte står i strid med skolans regler.

Men man glömmer ju, och är ibland oförsiktig. Låt oss därför upprepa några punkter.

- Ett lösenord är hemligt och personligt. Det får aldrig lånas ut.
- Kommersiell användning av datorresurserna är förbjudet.
- Olagligt eller pornografiskt material får inte lagras.
- Upphovsrättsligt skyddat material får inte distribueras.
- Anstötliga bilder ska inte förekomma på skärm eller utskrift.

Dessutom finns arbetsmiljöregler för labbsalarna anslagna i varje sal.

Personliga webbsidor ska innehålla texten "Detta är en personlig webbsida. Mer information." – en länk till högskolans webbreger – som kopieras från [KTHs webbplats](http://www.kth.se/gemensamt/disclaimer) (<http://www.kth.se/gemensamt/disclaimer>). På engelska används texten "This is a personal webpage. More information."

Reglerna är stränga och förbjuder bland annat politisk och religiös propaganda och annonsering för externa föreningar. Man får inte ha länkar till vad som helst och man får inte skämma ut högskolan i Sverige eller utomlands.

Bokningsläget för skolans labbsalar ses i KTHs schemagenerator, [TimeEdit](http://lokal2.timeedit.se/kth/) (<http://lokal2.timeedit.se/kth/>).³ Du har tillträde till datosalarna även utanför de tider din kurs bokat dem, men du måste lämna företräde ifall salen är bokad för en annan kurs. Ifall någon som bokat salen efterfrågar din plats, ska du *skyndsamt* spara ditt arbete och logga ut.

Läsanvisning

Det bästa sättet att lära sig använda ett UNIX-system, ifall man aldrig gjort det förut, är inte att läsa om det. Logga istället in på en dator och testa dig fram. Kör en webbläsare. Titta på filhanteraren. Se vilka inställningar som finns i skrivbordsmiljön – lägg in en snygg bakgrundsbild⁴. Sedan kan du titta igenom innehållsförteckningen för att se vad du behöver läsa.

Med det sagt, så är det även meningen att studenter på CSC även ska tillgodogöra sig mer grundläggande kunskap om datorsystem. Även om det går att använda ett modernt UNIX-system rent grafiskt (klicka sig runt i

³Gå till "Jag vill leta bland lokalerna...", välj datum och *Region*: "KTH-campus"; *Hus*: "Huvudbyggnaden"; *Lokal*typ: "Övriglokal".

⁴Högerklicka på skrivbordsytan, *Change Desktop Background*

Inledning

ett grafiskt gränssnitt för att starta program, flytta och ta bort filer), så bör man känna till hur filsystem är organiserade, hur en kommandorad fungerar och hur man kan använda fjärrinloggning för att komma åt datorsystem över Internet. Man bör även vara så bekant med de vanligaste operativsystemen att man hjälpligt kan använda även de system man inte personligen föredrar. Tänk på att även den som inte personligen kommer att programmera eller administrera datorsystem, troligen kommer att samarbeta med dem som gör det.

Följande typografiska konventioner används:

fast teckenbredd

Text i datorinteraktion, alltså text som du skriver in till datorn och den text datorn skriver ut. Där särskilt mellanslag även behöver framgå, skrivs `skrivs_texten_så_här`. Text som ska bytas mot annan, t.ex. användarnamn som ska bytas ut mot ditt eget, skrivs *så här*. Utlämnad text (som inte är relevant för exemplet) markeras med [...].

emfas

Text av särskild betydelse, såsom vanliga termer och begrepp; namn på program; text i engelsk översättning.

sans serif

Fönster, tangenter, menyrubriker, knappar och liknande.

Referenser

Ser ut [så här](#) ☺, med en ruta i marginalen som visar i vilken sektion ämnet tas upp. 3.1.1

Tangentkombinationer

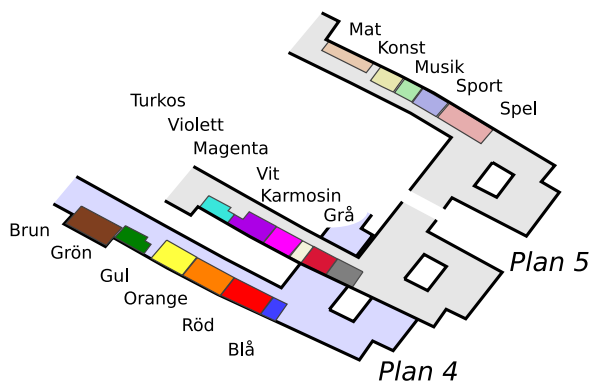
Skrivs som `Ctrl-X`, `Cmd-X`, etc. I vissa sammanhang (**Emacs** ☺) även som `C-x`. 8

I allmänhet används svenska beteckningar på de begrepp som tas upp, men med den engelska beteckningen inom parentes vid första omnämnandet för att du lättare ska känna igen dig när du läser dokumentation, som oftast är på engelska.

1 Inloggning


CSCs datorsalar finns på plan 4 och 5 av huvudbyggnaden. Notera dock att alla studenter inte har tillgång till samtliga datorsalar.

På Windows-datorerna loggar man in med samma konto¹, men det måste aktiveras separat om man går en kurs som kräver dessa system. Detta häfte behandlar inte Windows-systemen, annat än i hur de relaterar till UNIX-systemen (exempelvis hur man överför filer²).





E-huset (västra sidan)		D-huset (östra sidan)	
Plan 4	Plan 5	Plan 5	
Brun	20×Ubuntu	Mat	10×Win
Grön	8×Ubuntu	Konst	12×Mac OS X
Gul	20×Ubuntu	Musik	9×Mac OS X
Orange	21×Ubuntu	Sport	13×Ubuntu
Röd	21×Solaris	Spel	24×Ubuntu
Blå	(ej student)		
	Turkos		
	8×Mac OS X		
	Violett		
	16×Mac OS X		
	Magenta		
	14×Win		
	Vit		
	7×Win		
	Karmosin		
	14×Ubuntu		
	Grå		
	13×Ubuntu		

Alla CSCs system är *fleranvändarsystem*. För att använda dem måste man autentisera sig, dvs berätta för datorn vem man är (sitt användarnamn), och bekräfta det, vanligtvis med lösenord. På en arbetsstation görs detta när du loggar in.

Inloggnings-skärmar kan se olika ut. I vissa anger du användarnamn, och trycker sedan «**tab**» (märkt , ovanför Caps Lock) eller «**retur**» innan man skriver in lösenordet. I andra finns det två rutor, en för användarnamn och en för lösenord. Lösenordet visas aldrig när man skriver det. Ibland visas någon form av symbol för att representera varje tecken, i andra fall visas ingenting – det är alltså helt blankt.

Observera att du måste logga ut när du går ifrån datorn!

Om du lämnar datorn tillfälligt bör du låsa den, så att ingen kan använda datorn i ditt namn. Det är inte tillåtet att låsa datorn mer än **10 minuter**.

Vid problem att logga in, se respektive användarmiljö: [Gnome \(Ubuntu\)](#)  [3.1](#)
 eller [Mac OS X \(Finder\)](#)  [4.1](#)

1.1 Fillagring

De vanligaste programmen lagras lokalt, på den dator man loggar in, men din hemkatalog sparas i ett filsystem som heter AFS. Filer i AFS sparas på

¹Övergång till centraliserad KTH-inloggning är på gång, i samband med byte till Windows 7. (Troligen hösten 2010.)

²Fjärrinloggning  [7.5](#)

filserverar, och säkerhetskopieras dagligen (och varje vecka, månad, etc) för att minimera risken att man förlorar data.

[6.3.9](#) Alla hemkataloger har en kvot, typiskt 200 MB. Du kan se hur mycket du utnyttjar, vilka som har rättigheter, osv, med [kommandon](#) `ls -ld` och `ls -la`.

Säkerhet är till stor del baserad på ett protokoll som heter Kerberos. Du behöver sällan bekymra dig med detaljerna i detta, men ifall problem uppstår kan det vara bra att känna till kommandon som `kauth` eller `klist`. (Även om man inte själv använder dem kan de visa information som är bra att veta när man ber om hjälp.)

1.2 Datorsupport

CSCs salsdatorer administreras av Systemgruppen. Problem bör rapporteras till dem på system@csc.kth.se.

Även om de inte kan svara på kursrelaterade frågor, så kan de förhoppningsvis hjälpa till med andra datorfrågor.

[10](#)

För mer information, se [avsnittet om dokumentation](#), särskilt om [vem man kan fråga om hjälp](#).

[10.3](#)

2 Grafiskt gränssnitt (GUI)

Ett grafiskt användargränssnitt¹ används idag i allt från mobiltelefoner till bankomater. När det gäller datorer, brukar de flesta GUI:n innehålla ungefär samma möjligheter. Program visas i ett eller flera fönster, som har en titelrad med knappar för att maximera, minimera och stänga fönstret (och kanske programmet). Man aktiverar ett fönster genom att klicka i det (eller på ramen), och det aktiva fönstret (programmet) är det som har fokus, dvs det program som tar emot text som skrivs på tangentbordet, musrörelser etc.

En skrivbordsmiljö är det program som bestämmer hur detta GUI ska se ut – färgen på fönstrens ramar, var minimera/maximera-knapparna ska sitta, ikoner, paneler, verktygsfält för att starta program och så vidare. Skrivbordsmiljön är oftast hårt knuten till en filhanterare, och innehåller även funktionalitet från denna. Exempelvis kan man oftast dra-och-släppa filer direkt från skrivbordet till ett öppet program, på samma sätt som man i filhanteraren drar filer mellan olika kataloger.

2.1 Mus och tangentbord

Vad kan man säga om mus och tangentbord? Fungerar inte de likadant under UNIX som i andra operativsystem? Jo, det gör de. Mer eller mindre. Men man har en del andra traditioner, och det finns ofta fler val för att ändra tangentbordsinställningarna.

2.1.1 Mus – alla tre knappar används

Arbetsstationer med UNIX har länge haft tre-knappars möss, även när exempelvis Windows bara hade två knappar. Därför används den tredje (mittensta) musknappen mer än i andra OS.

Den första (vänstra) knappen används på sedvanligt vis, liksom den andra (högra), som oftast ger en kontextmeny. Dessutom kan ett klick på mittenknappen (eller scrollhjulet) användas för att klistra in text, som man först markerat med första knappen. Detta gör det lätt att kopiera in ett ord som "Clethrionomys"² genom att dubbelklicka för att markera ordet, och sedan mittenklicka i sitt textdokument för att klistra in det.

Vissa webbläsare använder istället mittenklick för att öppna en länk i ett nytt fönster, eller en ny flik (*tab*).

2.1.2 Tangentbord

Alla moderna tangentbord har tangenter för att skriva versaler (*Shift*) och en kontrolltangenter (*Control*), men andra modifieringstangenter varierar med olika tangentbord och/eller OS. Även om termerna för de så kallade *Super*-tangenterna (*Command*, *Win* och *Meta*; se nedan) kan variera, så fungerar de flesta kortkommandon ungefär likadant.

Det krångligaste är förmodligen att veta vilka termer man ska söka efter i dokumentationen. I UNIX-världen kan detta ibland vara krångligt; eftersom få PC-tangentbord haft en *Meta*-tangenter, så har man istället använt *Alt*-tangenter för det syftet. Därför trycker man `Alt-D` för att ta bort det hela ord som står efter markören, men det benäms `Meta-D` eller `M-d` i dokumentationen.

För detaljer om tangentborden, se figur 1 på sid 9. Kortfattat:

¹Vanligen kallat GUI, från engelskans *Graphical User Interface*

²för alla som skriver om långsvansade skogssorkar

Tekniskt:	Command \approx Windows \approx Meta Alt = Option	(Super-tangenter) (Alt-tangenter)
Praktiskt:	Command \approx Control Alt \approx Meta	(Genvägar) (i program)

Det är inte så förvirrande som det verkar, men det kan vara bra att lära sig några termer.

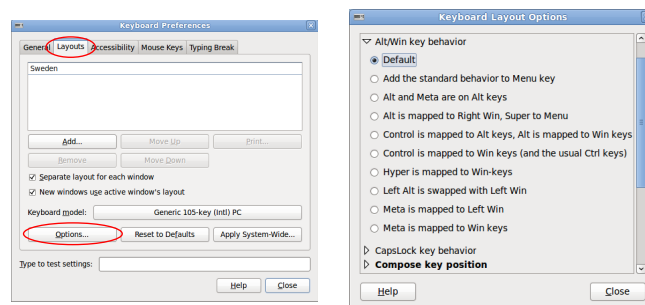
Mac Ett Mac-tangentbord har en kommandotangent (*Command*, *Cmd*) och en alt-tangent (*Option*, *Opt* eller *Alt*). Kortkommandon använder oftast `Cmd-«tangent»`.

Windows "Windows-tangentbord³" har *Control* (*Ctrl*), *Win*, *Alt*, *Alt Gr* och *Meny* (eng. *Menu*, med en stiliserad kontextmeny⁴ på). Men *Win* och *Alt*-tangenterna är placerade tvärt emot Mac-tangentbordets. Kortkommandon använder vanligen `Ctrl-«tangent»`.

Solaris (Sun) UNIX har traditionellt haft fler modifieringstangenter än andra OS. Det har funnits tangentbord med upp till sju olika tecken tillgängliga på varje tangent.

Ett Sun-tangentbord innehåller, förutom de särskilda genvägstangenterna till vänster om de vanliga alfanumeriska tecknen, tangenterna *Control*, *Alt*, *Meta*, *Compose* och *Alt Graph*. Kortkommandon använder oftast *Control*, men även *Meta* är vanligt. Java Desktop System (JDS), som finns på Solaris-maskinerna, använder dock *Alt*.

Ubuntu (Linux) Linux använder, när man har ett Windows-tangentbord, tangenten *Alt* som *Meta*. *Win*-tangenten (kallad *Super*) används ganska sällan. Men det finns möjligheter att själv ställa in funktioner, under **System** → **Preferences** → **Keyboard** → **Layouts** → **Options**.



Precis som vanligt så används *Alt Gr*-tangenten⁵ för att skriva vissa tecken i svensk eller annan internationell tangentbordslayout. Till exempel, så ger tangenten "2"

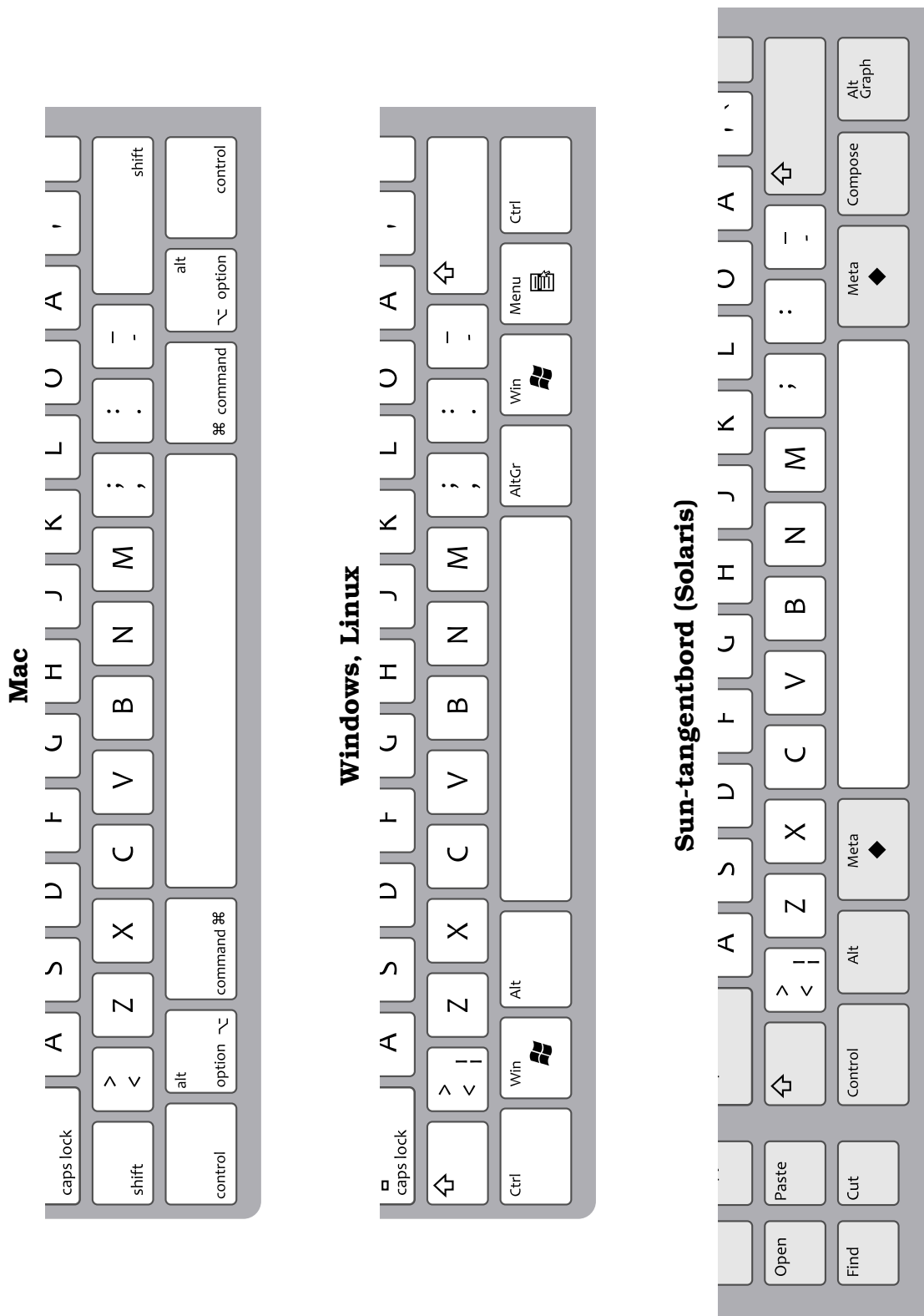
Modifieringstangent:	(inget)	shift	alt gr	shift + alt gr
Tecken:	2	"	@	² (upphöjd)

Dessutom kan UNIX, även om inte alla tangentbord har det, använda en *Compose*-tangent. Det är en s.k. "död" tangent, ungefär som den tangent som används för att skriva akut och grav accent i "á" och "à", eller tangenten för ~ (tilde) i "ñ".

³Formellt ett 104 eller 105 tangents IBM PC-tangentbord.

⁴Context menu; kallas även *popup-menu*, och kan oftast fås genom att högerklicka

⁵Kallas även *Alt Graph*, *Mode Shift* eller *ISO Level3 Shift*



Figur 1: Jämförelse mellan tangentbord

Med en *Compose*-tangente kan man t.ex. skriva *Compose a e* för att skriva ett "æ", eller *Compose , c* för "ç".

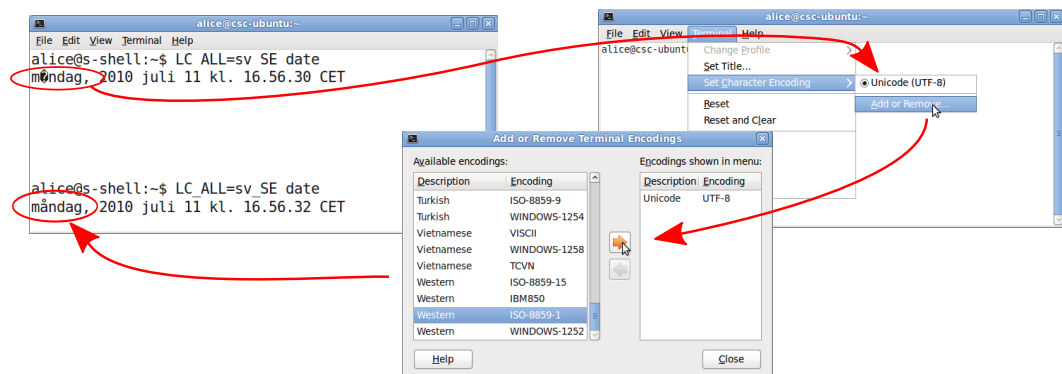
Ett vanligt val är att (enligt ovan) ställa in en av *Win*-tangenterna, eller *Caps Lock* till att vara *Compose*-tangente.

2.2 Teckenkodning (UTF-8 eller Latin-1)

Hur man kodar text, alltså vilka binära sekvenser som representerar bokstäver, kan variera mellan olika system. Oftast kan program som Emacs känna igen vilket sätt som använts, och automatiskt konvertera text ifall det behövs. Men ibland fungerar inte det, och då kan du behöva ställa in det manuellt.

På CSCs Solaris-system användes i många år en teckenkodning som heter *latin-1* (mer formellt *ISO-8859-1*). Nu använder både Mac OS X och Ubuntu *Unicode*, sparad med teckenkodningen *UTF-8*.

Mest märkbart är skillnaden ifall du har ditt terminalfönster felinställt när du loggar in på en Solaris-maskin. Då kan det göra att alla ÅÅO ersätts med något konstigt tecken, tills du byter.



Ifall man vill konvertera innehållet i en textfil, så kan man istället för att öppna den i en texteditor och spara om med en annan teckenkodning använda [kommandoradsprogrammet](#) `iconv`:

```
> iconv -f iso8859-1 -t utf-8 < original.txt > ny.txt
```

Konvertering fungerar dessvärre inte alltid. Ifall texten innehåller tecken som inte kan konverteras, så får du ett felmeddelande:

```
> iconv -f utf-8 -t iso8859-1 < cyrillic.txt
iconv: illegal input sequence at position 3
```

Tyvärr finns det många snarlika teckenkodningar, även om de flesta program går över till att använda antingen UTF-8 eller latin-1. Några man kan behöva skilja på:

ASCII Äldre amerikansk teckentabell, innehåller inte mycket mer än engelska alfabetet. De flesta modernare teckentabeller baseras på ASCII (och innehåller det som en delmängd).

ISO-8859-1, Latin-1 Teckentabell med "latinska" tecken, dvs tecken som används i (väst)europiska språk. Innehåller också symboler för hårt blanksteg, gradsymbol (°), etc.

2.2 Teckenkodning (*UTF-8* eller *Latin-1*)

Windows-1252 En utökning av Latin-1, använd i MS Windows, med exempelvis typografiska citattecken tillagda.

ISO-8859-15 Förändring av latin-1, för att byta ut några ovanliga tecken ($\frac{1}{4}$, $\frac{3}{4}$, etc) mot tecken som används i flera europeiska språk (€, œ, etc).

Unicode Unicode innehåller de flesta vanliga teckenuppsättningar⁶, inklusive kinesiska, arabiska, katakana, etc, och även runskrift (fupark) och många matematiska, tekniska och musikaliska symboler.⁷ UTF-8 används för att spara dessa på ett kompakt binärt format.

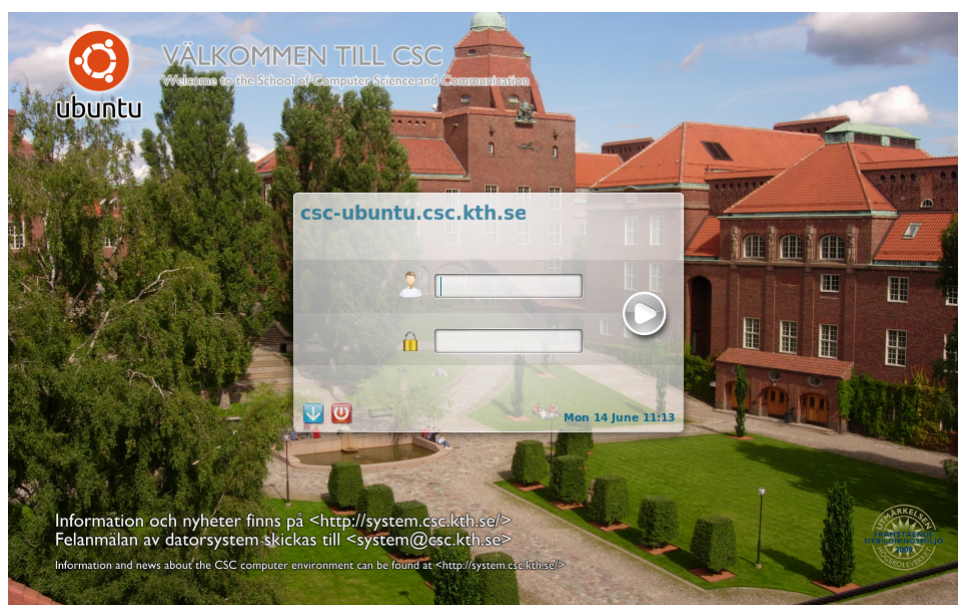
UTF-8 8-bitars *Unicode Transformation Format*. Kan representera alla Unicode-tecken, och är bakåtkompatibelt med ASCII, så långt detta nu är möjligt.

⁶Över 100 000 tecken, i senaste versionen.

⁷Ifall du kan använda alla tecken beror också på ifall mjukvaran och/eller typsnitten stödjer dem. Vilket inte alltid är fallet.

3 Ubuntu

3.1 Inloggning



- 10.4** För att logga in använder du ditt CSC-konto. Du kan alltså *inte* använda det **centrala KTH.SE-kontot** [↗](#) (som används på ex *Mina Sidor*). CSC-kontot har (för de flesta) samma användarnamn som KTH.SE-kontot, men inte samma lösenord.¹ Byter du det ena lösenordet, så byts inte det andra.

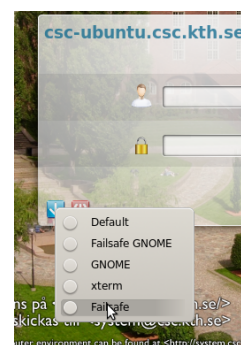
3.1.1 Felsäker inloggning (vid problem)

Ifall du har problem att logga in kan det hjälpa att logga in med *felsäker session*. Om du klickar på den blå pilen nere till vänster, så kan du välja **Failsafe GNOME** eller **Failsafe**. Det första ger dig default-inställningar för fönsterhanteraren (Gnome), men i övrigt kan du använda systemet normalt. Det andra gör att du bara får upp ett terminalfönster. Det kan vara användbart ifall du helt vill rensa bort dina Gnome-inställningar – annars sparas de på nytt när du loggar ut.

Du kan även använda det ifall man har så fullt på kontot att det inte går att logga in². Du måste känna till (eller slå upp) hur man använder kommandoradsmiljön.

10.5.4

För att återställa standardinställningarna, se avsnittet om att [återställa inställningar](#) [↗](#).



3.1.2 Om du inte kan logga ut

Om du har problem att logga ut, eller ifall någon har låst skärmen på en dator *när din kurs har bokad salen*, så kan du använda **Ctrl-Alt-backsteg** för att döda X-servern (den startar om med en ny login-skärm).

Observera att detta dödar alla program som körs, och **öppna filer kan försvinna**. Det ska inte missbrukas för att logga ut andra.

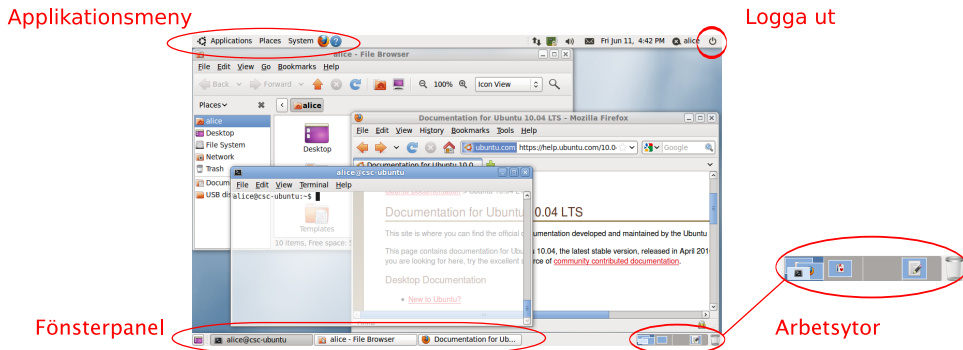
¹Även om det är dåligt ur ett säkerhetsperspektiv, så kan man sätta samma lösenord på bägge konton. Vi råder dig att *inte göra det!*

²Din hemkatalogs storlek begränsas av en [kvot](#) [↗](#) **7.2.2**

3.2 Gnome Desktop

3.2 Gnome Desktop

Den fönstermiljö som möter dig när du loggar in kallas *Gnome*. Nästan alla delar av fönstermiljön går att ändra, ifall du så önskar, men defaultinställningarna ser ut ungefär så här:



De viktigaste delarna är:

Applikations- och systemmenyer Här hittar du de vanligaste programmen (*Applications*), katalogerna (*Places*) och systeminställningarna (*System*).

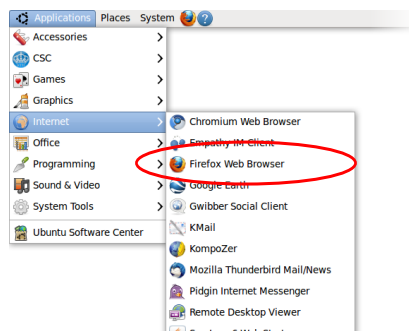
Fönsterpanel De program du har öppna visas här.

Arbetsytor Du kan öppna olika program i olika skrivbordsytor, exempelvis för att kunna bläddra fram och tillbaka mellan ett program och dokumentation/labbdyelse e dyl.

Logga ut Du loggar ut genom att använda ikonen uppe till höger.³

3.2.1 Starta applikationer (program)

De flesta program finns i undermenyer till **Applications**, exempelvis så finns Firefox under **Applications** → **Internet** → **Firefox Web Browser**.

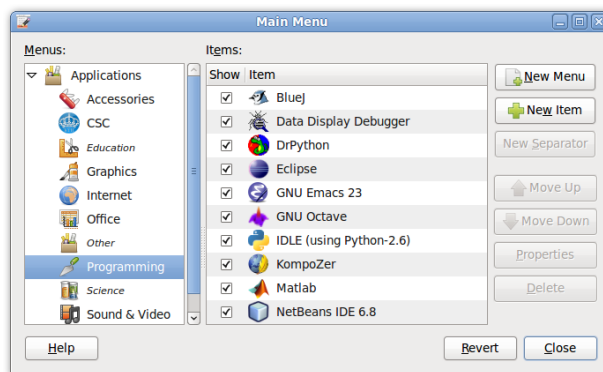


Eftersom CSCs datorer används av många olika kurser, så är ett stort antal program installerade. Detta kan göra det svårt att hitta i programmenyn. Ifall du vill ta bort några av de program du inte använder, eller lägga till några du själv föredrar, så går du till **System** → **Preferences** → **Main Menu**.

Här kan du sedan ta bort eller lägga till program från menyerna.⁴

³Det går att ta bort den panel ikonen ligger i; i så fall får man en meny för utloggning under *System*.

⁴Menyn *Places* styrs istället av vilka bokmärken du har i filhanteraren, Nautilus.



Ifall du ofta vill komma åt ett visst program, så kan du även lägga till en ikon för dem på skrivbordet genom att högerklicka och välja **Add this launcher to desktop**. Ikonen går sedan att flytta dit man vill.

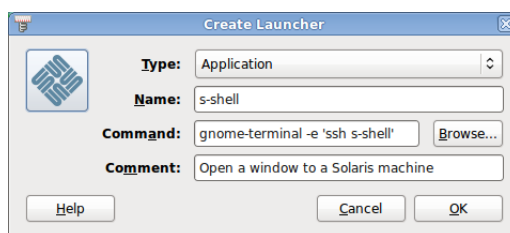
Du kan även att göra helt egna *Launchers*, dvs ikoner på skrivbordet eller vid applikationsmenyn. Högerklicka på skrivbordet, välj **Create Launcher...**, och skriv in:


Type: Application

Name: Det namn som ska synas vid ikonerna, som ett filnamn

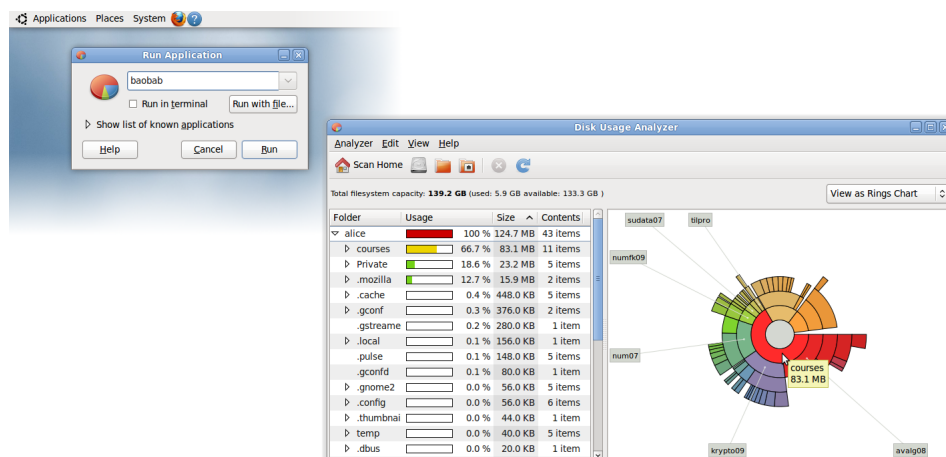
Command: Kommandot som ska utföras

Comment: Ifall ikonerna ligger i en meny, visas denna text när man håller musen över.




7.5 Du kan också välja någon lämplig ikon. Denna launcher skulle öppna en terminal för **fjärrinloggning**  till en Solaris-maskin, så en gammal Sun-logo⁵ verkar passa.

Ifall du vet namnet på applikationen så kan du köra den genom att trycka **Alt+F2**, och skriva in programmets namn, t.ex. *baobab*, som även går att nå under **Application** → **Accessories** → **Disk Usage Analyzer**.



⁵Sun är nu ett dotterbolag till Oracle

3.3 Internet och kommunikation

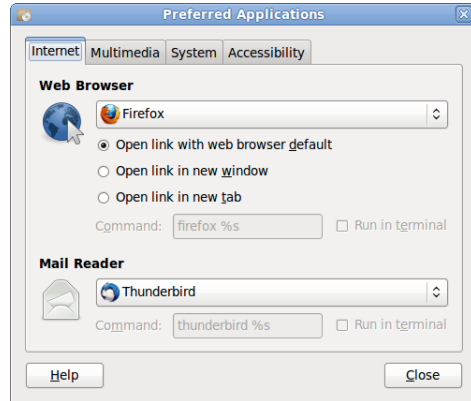
Notera dock att *baobab* i skrivande stund inte förstår sig på [AFS-kvoter](#) . [7.2.2](#)
Den kan därför inte visa hur mycket plats du har ledigt på kontot.

3.3 Internet och kommunikation

Den vanligaste webbläsaren är Firefox, men även exempelvis Chromium finns installerade.

För den som inte vill använda sig av Web-mail (att läsa sin e-post i en webbläsare), så brukar Thunderbird vara populär. Det finns dock även den mer Outlook-liknande Evolution som förutom e-post hanterar kalender-funktioner.

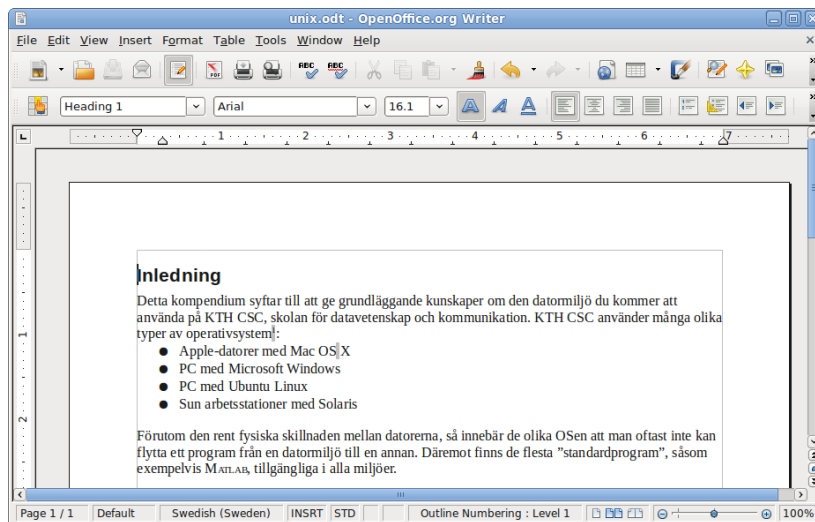
För att automatiskt starta rätt program när man klickar på länkar e dyl, så kan man gå till **System** → **Preferences** → **Preferred Applications**.



3.4 Ordbehandling och kalkyl

Kontorspaketet OpenOffice kan öppna de flesta ordbehandlings- och kalkylbladsformat.

Ifall du ska flytta dina dokument mellan många olika miljöer, kan det vara bäst att använda RTF-format, eller liknande.⁶

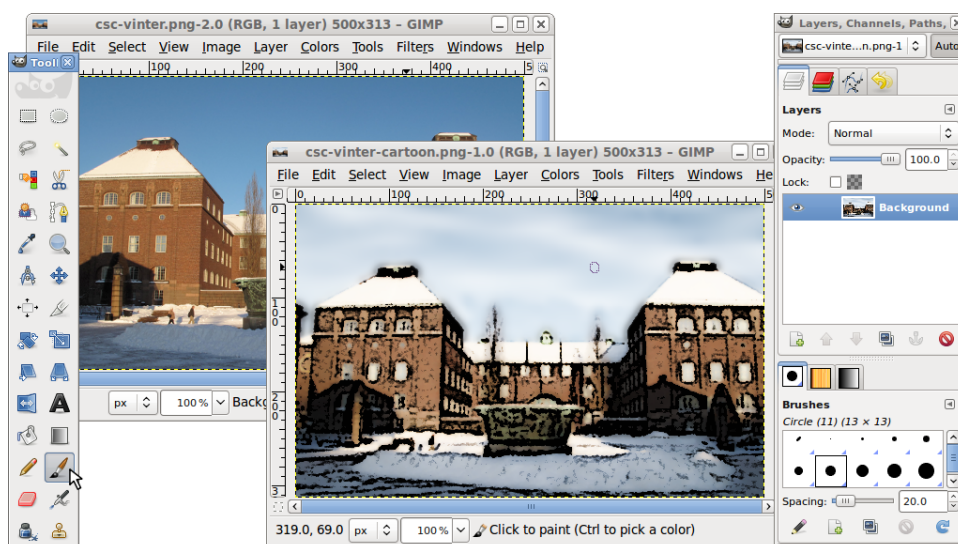


3.5 Grafik

Det finns många olika program som kan utföra vanliga uppgifter. Bilder öppnas normalt i *Image Viewer* (eller "Eog", *Eye of Gnome*), som kan rotera bilder och spara dem i olika format.

Ifall du vill redigera bilder, så är förmodligen *GIMP* (*GNU Image Manipulation Program*) det mest kraftfulla program, som dock ofta kritiserats för sitt användargränssnitt.

⁶Äldre versioner av Microsoft Office kan annars ha problem med de nyare *Open XML*-formaten (även kallat *DOCX*) som är default i senaste versionerna av MS Office. (Openoffice kan öppna de flesta versioner av dem.)



Andra grafikprogram inkluderar *Dia* (används för att skapa diagram) och *Blender* (verktyg för 3D-grafik).


För vektorgrafik⁷ finns programmet *Inkscape*, som har som mål att följa standardformat som XML, SVG och CSS.

För *desktop publishing* (datorstödd layout av trycksaker) så finns programmet *Scribus*.

I praktiken är dock industristandarden för grafik Adobes *Creative Suite*. Dess program (*Photoshop*, *Illustrator*, *InDesign*, mfl) finns inte för Linux då de inte görs för andra operativsystem än Windows och Mac OS X. De finns dock tillgängliga på de flesta av CSCs Mac och Windows-datorer, om än med begränsat antal licenser.

3.6 Texteditor

Det är viktigt att förstå skillnaden mellan *ren text* (oformaterad text, eng. *plain text*) och *formaterad text*. Oformaterad text innehåller ingen typ av formatering – kursiv eller fetstil, typsnitt, tabeller etc. Det enda sättet att "formatera" texten är att använda vanliga mellanslag, och tecken som #*_ _ för att markera ord. Programmering görs oftast i någon form av ren text, liksom konfigurationsfiler.

8 Vilken texteditor man använder för att skriva text spelar mindre roll, även om en del programmerare har starka åsikter om vilken som är "bäst". De två traditionella är *Emacs*  och *vi*. För att användas i programmering bör en editor ha automatisk indentering⁸ och syntaxfärgning (eng *syntax highlighting*)⁹

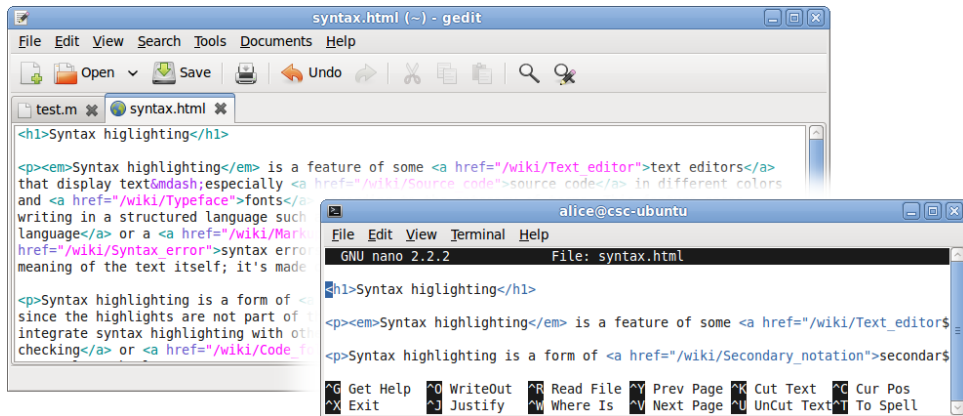
Ubuntus texteditor (**Application** → **Accessories** → **Text Editor**) heter *gedit*.

⁷Vektorgrafik bygger upp bilder med linjer och kurvor, vilket gör att man lättare kan skala om det, utan att förlora detaljer.

⁸Automatisk justering av vänstermarginalen med mellanslag, för att markera var avsnitt av koden börjar och slutar

⁹Markera med färger, typsnitt e dyl för att göra koden mer lättläst. Notera att detta inte sparas i filen – texten är oformaterad. Det läggs till av editorn.

3.7 Flyttbara media (USB-minnen etc)

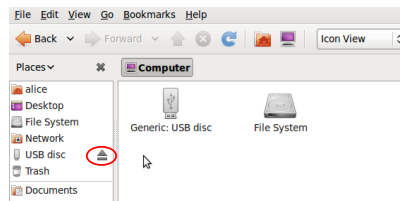


Det finns dock även andra. Programmerar man i en utvecklingsmiljö som *Eclipse* finns det oftast en integrerad editor. Arbetar man (pga fjärrinloggning eller liknande) i en ren terminalmiljö finns *nano*. (På Solaris är *mg* vanligare.)

3.7 Flyttbara media (USB-minnen etc)

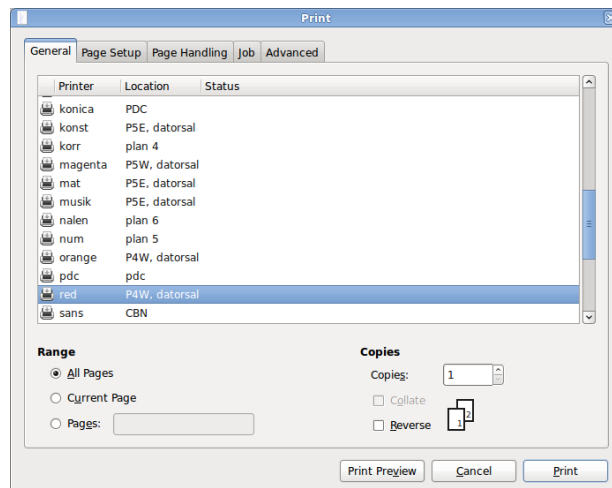
Ubuntu känner igen de flesta flyttbara media. Ifall du stoppar i en USB-sticka så bör den dyka upp under **Places**.

Tänk dock på att innan du tar bort enheten, så bör du avmontera den (motsvarar ungefär att i Windows klicka på *Safely remove hardware*). Du avmonterar den genom att klicka på den ikon som syns bredvid enheten i filhanteraren.

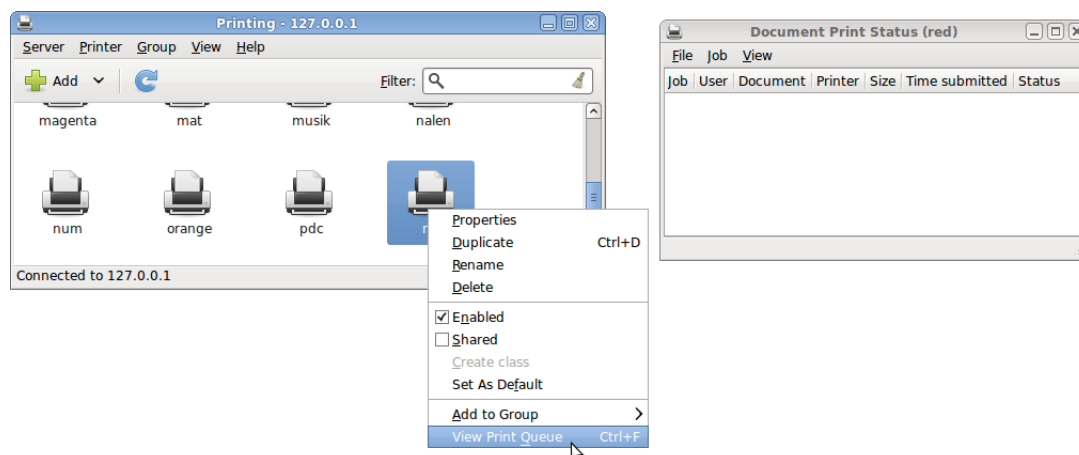


3.7.1 Utskrifter

Gnomes utskriftsdialog låter dig välja skrivare, och har även möjligheter (på de andra flikarna) att ställa in saker som *duplex* (dubbelsidig utskrift), vilket fack som används, etc. Vilka val som finns, beror givetvis på skrivaren.



Du kan se utskriftskön om du går till **System** → **Administration** → **Printing**, och där väljer **View Print Queue** genom att högerklicka på en skrivare.



3.8 \LaTeX

\LaTeX är ett typsättningsprogram som använder en textfil skriven i ett märkspråk¹⁰ för layoutbeskrivning och skapar ett typsatt dokument. Det är populärt inom områden som har att göra med teknik och matematik, då det har kraftfulla verktyg för att typsätta matematiska formler. Det innehåller dokumentmallar för artiklar, böcker, brev, presentationer, m m.

```
Om  $u = f(x)$  och  $x = g(\mathbf{t}) = g(t_1, \dots, t_q)$  så är
\begin{displaymath}
\frac{\partial u}{\partial t_j} = \frac{du}{dx} \cdot \frac{\partial x}{\partial t_j} \quad \forall j \in \{1, \dots, q\}
\end{displaymath}
```

Om $u = f(x)$ och $x = g(\mathbf{t}) = g(t_1, \dots, t_q)$ så är

$$\frac{\partial u}{\partial t_j} = \frac{du}{dx} \cdot \frac{\partial x}{\partial t_j} \quad \forall j \in \{1, \dots, q\}$$

Det kan se krångligt ut att skriva, men det går (när du är van) mycket snabbare att redigera komplicerade formler än de alternativ som brukar finnas i ordbehandlare.

\LaTeX är gjort för att följa lämpliga regler för typografi: lagom radavstånd för att vara lättläst, lämpliga storlekar för rubriker, etc. Detta gör att du kan koncentrera dig på att skriva texten och låta \LaTeX sköta utseendet.

En introduktion till \LaTeX finns i: [The Not So Short Introduction to \$\LaTeX\$ 2ε](http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf) (<http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf>)

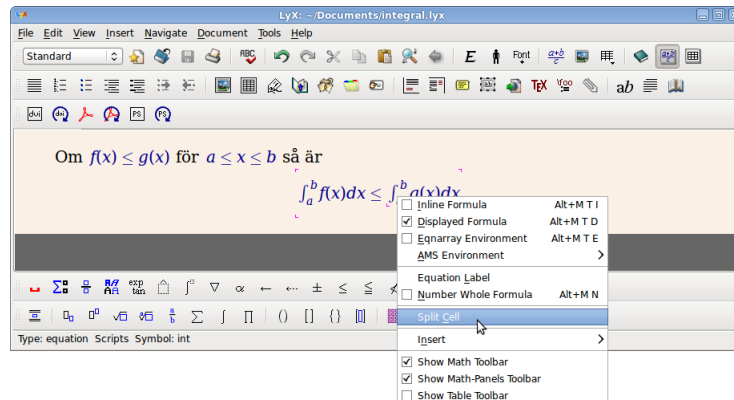
Den är (som namnet antyder) inte så kort, men delar av den kan man hoppa över, ifall man inte behöver skriva på ryska, thai eller annat främmande språk.

¹⁰Ett märkspråk (eng *markup language*) är en slags specialiserat programspråk med särskild text, som inte syns utan används för att förändra hur övrigt innehåll presenteras. Dessa kan vara både strukturella (markera paragrafer, kapitel etc), semantiska (beskriva relationer mellan textdelar: referenser, citat, etc).

Exempel är, förutom \LaTeX och andra varianter av \TeX , HTML, XML och RTF. Ordbehandlares dokumentformat innehåller oftast samma typ av direktiv, men som binärkod istället för att vara utskrivna som (mer eller mindre) klartext.

3.9 Stavningskontroll: spell, stava

För den som inte vill lära sig skriva \LaTeX -kod, så finns även en grafisk editor baserad på \LaTeX , \LyX :



Även om det ser ut som en vanlig ordbehandlare, så är arbetssättet och filosofin bakom annorlunda. Det finns en bra (engelsk) introduktion under **Help**-menyn.

3.9 Stavningskontroll: spell, stava

Kontorsverktyg (ordbehandlare o dyl) har oftast inbyggda verktyg för stavningskontroll. Men eftersom UNIX traditionellt sett använt filer med ren text, och hanterat dem via [kommandoradsverktyg](#) 6, så finns specialiserade verktyg även för detta. Den etablerade stavningskontrollen i UNIX är *spell*, och det finns ett par olika versioner, exempelvis *ispell* och *aspell*.

spell söker igenom indata efter ord som inte finns i dess ordlista eller (vilket är vanligare idag) ord som inte går att konstruera med en språkalgorithm. Felaktiga ord skrivs på *standard output*.

```
> cat spell_check.txt
Eye have a spelling chequer,
It came with my Pea Sea.
It plane lee marks four my revue
Miss Steaks I can knot sea.
> spell spell_check.txt
chequer
```

Som du ser kan man behöva använda väljaren `-b` för att *spell* inte ska flagga ord med brittisk stavning.

Det går också att anropa med en [pipeline](#) 7.1.5, för att snabbt kontrollera ett ord:

```
> echo parallell or paralell | spell
parallell
```

Det finns även andra sätt att anropa *spell*, t ex genom `M-x spell-word` i texteditorn [Emacs](#) 8.

stava är en svensk stavningskontroll, som kan användas ungefär som *spell*:

```
> echo parallell eller paralell | stava
paralell
```

stava är ett lokalt program på CSC. Mer information, och en länk till dokumentationen, finns ifall du anropar med väljaren `-?`.

4 Mac OS X

4.1 Inloggning

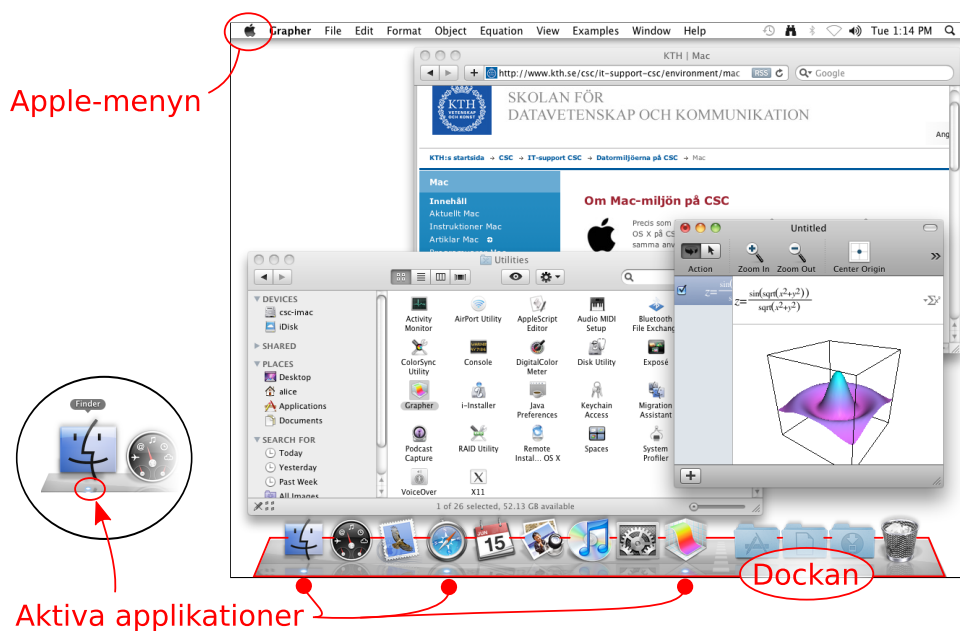


Du loggar in på vanligt sätt, med samma konto som på övriga UNIX-system. Ifall du får problem med inloggningen, eller ifall systemet känns ovanligt slött, så kan det bero på att ditt konto är **fullt** ☹. Särskilt katalogen Library kan lätt bli full på Mac:ar.

7.2.2

4.2 FINDER

De vanligaste applikationerna går att nå nere i "dockan", längst nere på skrivbordet. Där kan man även lägga genvägar till de applikationer man själv vill använda. Övriga applikationer når man genom att öppna Finder, och välja **Go** → **Applications**, eller kortkommandot **Cmd-Shift-A**.



Standardprogram på CSC – MATLAB, Firefox, Thunderbird – finns installerade, och fungerar i stort sett likadant som på övriga operativsystem.

4.3 Internet

Dessutom finns givetvis de program som följer med OS X, såsom iTunes, Preview, osv.

4.2.1 Flyttbara media (USB-minnen etc)

En USB-enhet dyker på en Mac upp både på skrivbordet och bland *Devices* i Finder. För att ta bort enheten, så drar du den till papperskorgen nere på dockan. Du kan se att ikonerna för papperskorgen ändras till en Eject-symbol.



4.2.2 Keychain

Keychain är ett program som ingår i OS X, för att hålla reda på de olika lösenord man använder. Förutom det lösen du loggar in med, så brukar de flesta behöva använda olika lösenord på webbsidor, nätverksanslutningar, trådlösa nätverk, krypterade filer, etc, etc. I *Keychain* sparas dessa lösenord (krypterade), och applikationer kan ta fram dem när de behövs.

För att hantera din nyckelring använder du applikationen *Keychain Access* som finns under **Applications** → **Utilities**.

Om ett program är inställt på att spara dem, så kommer OS X att fråga ifall du vill spara de lösenord du använder:



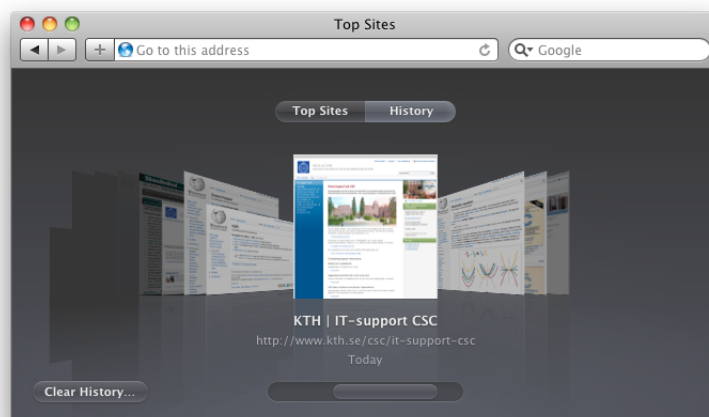
Ifall *Keychain* börjar fråga om lösenord, så kan dess inställningar ha blivit korrupta¹. Förhoppningsvis kan du återställa lösenordet till samma som du använder för att logga in. I värsta fall kan du behöva ta bort din nyckelring, se systemgruppens webbsidor:

- [Hur man tar bort sin nyckelring](http://www.kth.se/dokumentation/it-support-csc/environment/mac/instructions/keychainremove) (<http://www.kth.se/dokumentation/it-support-csc/environment/mac/instructions/keychainremove>)
- [Hur man ändrar lösenordet på sin nyckelring](http://www.kth.se/dokumentation/it-support-csc/environment/mac/instructions/changekeypasswd) (<http://www.kth.se/dokumentation/it-support-csc/environment/mac/instructions/changekeypasswd>)

4.3 Internet

OS X:s webbläsare heter Safari, och ligger normalt i dockan.

¹Detta beror oftast på att man har [fullt på kontot](#) [7.2.2](#).



4.4 Ordbehandling

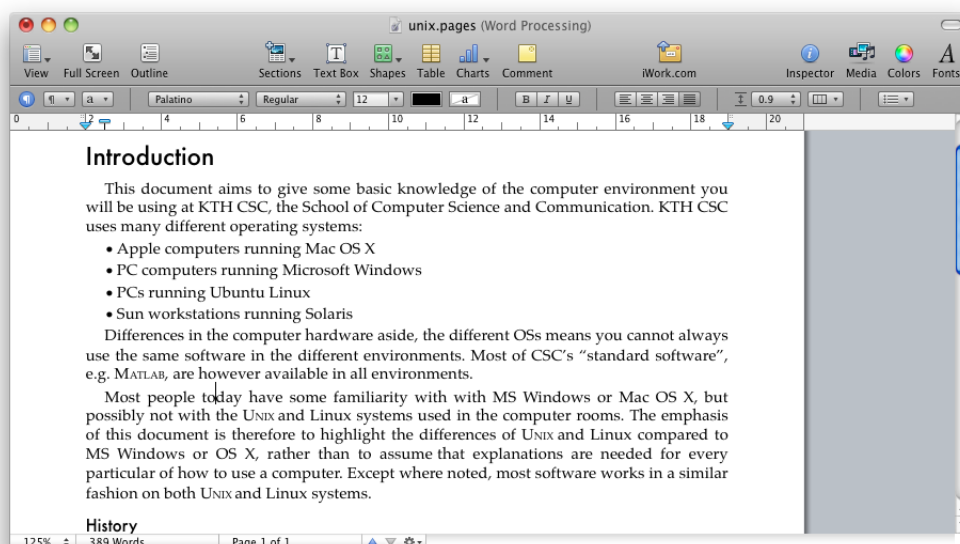
4.4.1 iWork

Apples kontorssvit heter iWork. Dess Office-program är förmodligen de som är mest integrerade i OS X. De program som ingår är:

Pages för ordbehandling och layout.

Numbers för kalkylblad (beräkningar, datatabeller, grafer, etc).

Keynote för presentationer.



4.4.2 MS Office 2008

Alla CSCs mac:ar har dessutom *MS Office 2008* installerat. Det har i stort sett samma program som Windows-versionen:

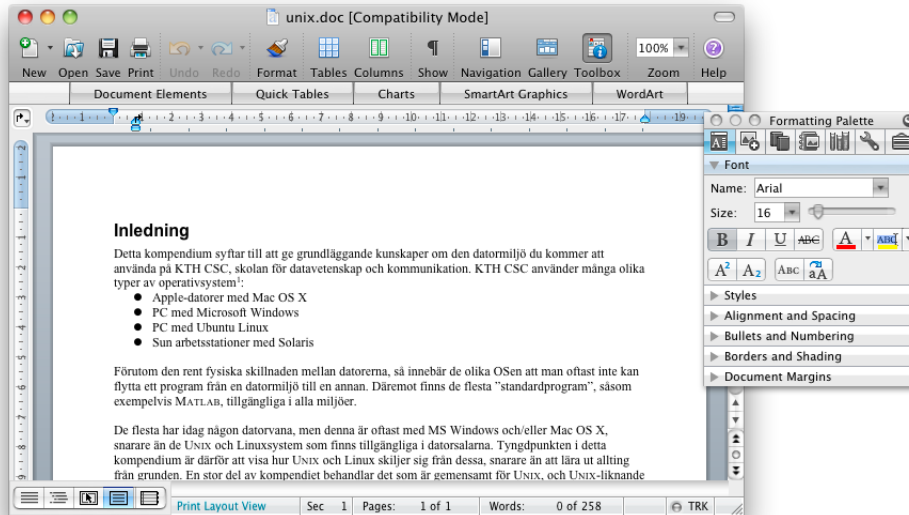
Word för ordbehandling

Excel för kalkylblad

4.5 Grafik

Powerpoint för presentationer

Det ingår också *Entourage*, för epost- och kalenderfunktioner (ersätter *MS Outlook*, som inte finns för OS X). *Entourage* är dock inte installerat på CSCs datorer².



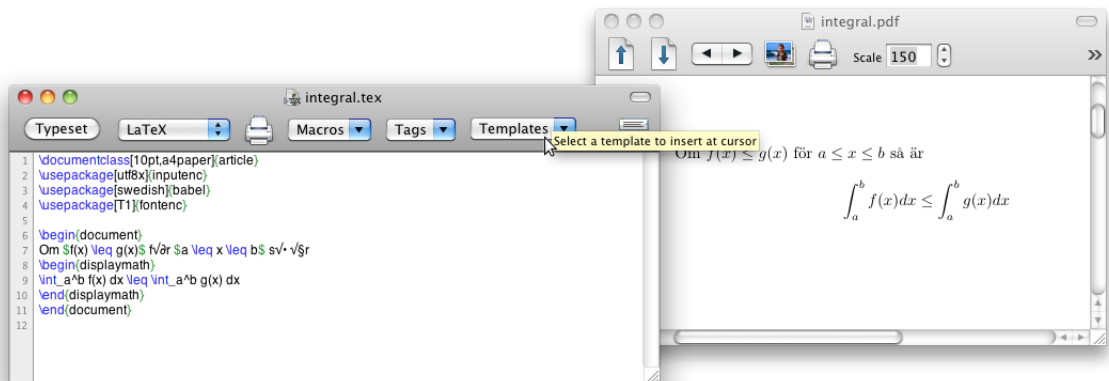
4.4.3 Ren text; \LaTeX

Man kan också använda *TextEdit* för att skriva enklare texter (och oformaterad text).

Mac:arna kör, liksom Ubuntu-maskinerna, TeXLive, så det ska fungera i stort sett likadant att använda [kommandoradsprogrammen för \$\text{\LaTeX}\$](#) . 3.8

Det finns även en grafisk miljö anpassad för att skriva \LaTeX -kod, *TeXShop*.

Programmet har ett fönster för att skriva koden, och ett för att se resultatet.



4.5 Grafik

4.5.1 Adobe-sviten: CS5

Adobe Creative Suite, version 5, eller CS5, är installerat på alla datorer. Detaljerna i hur dessa program fungerar är minst sagt utanför vad detta

²Det sätt programmet sparar data fungerar inte bra med en hemkatalog i AFS.

häfte kan ta upp, men programmen utgör standard inom grafiska tillämpningar. Dokumentation finns tillgänglig både på Internet och i bokform.

De vanligaste bland de många program som ingår är:

Photoshop för att redigera foton, eller andra bilder som bygger på rastergrafik ("pixelgrafik").

Illustrator för att redigera vektorgrafik.

InDesign för att skapa trycksaker (planscher, broschyrer, böcker, ...) eller andra. ("Desktop publishing".)

4.5.2 Andra program (urval)

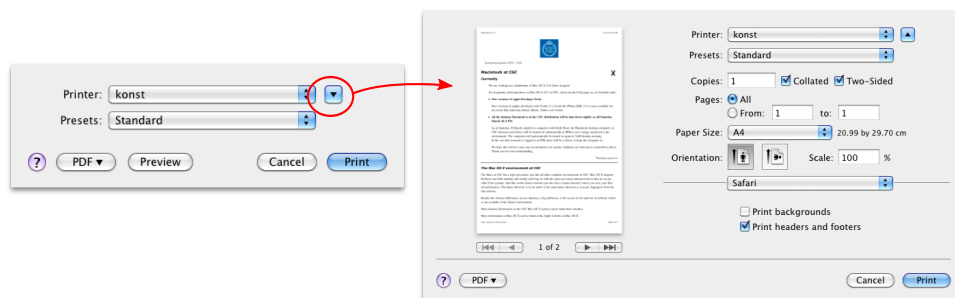
Autodesk Maya används för att skapa 3D-animationer, 3D-modeller, simuleringar, visuella effekter och liknande. Programmet används inom film- och TV-industrin, men också för att göra datorspel, arkitekturritningar och annan design.

Final Cut Pro används för att producera och redigera filmer. Har använts för att göra många stora Hollywood-filmer, men är också populärt bland mindre filmmakare.

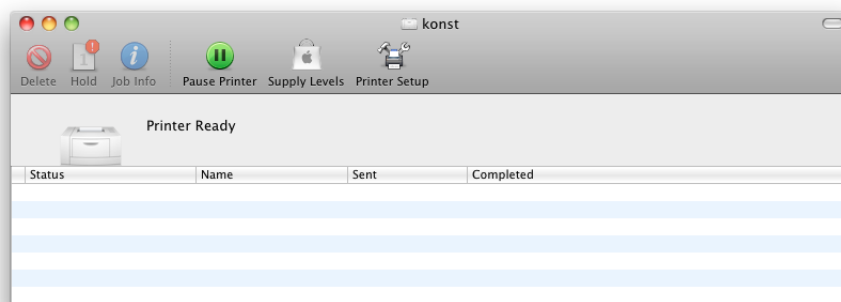
Adobe Dreamweaver³ används för utveckling av websidor. Kan användas för att skapa och underhålla hela webplatser, med inbyggda verktyg för filöverföring. Har både möjlighet att redigera websidor i grafiskt läge (s k WYSIWYG, *what you see is what you get*), och att skriva koden direkt.

4.5.3 Utskrifter

I de flesta program används OS X:s utskriftsdialog. Man kan nöja sig med att välja skrivare, eller klicka på popup-menyn för utökade inställningar.



Du kan se utskriftskön genom att gå till *System Settings*, och välja *Print & Fax*.



³Tidigare *Macromedia Dreamweaver*.

5 Solaris

5.1 Inloggning

Vid inloggningsrutan på Solaris-system kan du välja olika fönstermiljöer. *Java Desktop System* (JDS) är baserad på Gnome, men uppbyggd kring ett något mer Windows-liknande utseende. Dessutom finns *Common Desktop Environment* (CDE), som för många år sedan var default-miljö på CSC, och fortfarande används av en del användare.



5.1.1 Felsäker inloggning (vid problem)

Som du kan se i inloggningsskärmen ovan, så har du i Solaris (liksom [Ubuntu](#)) möjligheten att logga in med felsäker inloggning. Då öppnas endast ett terminalfönster, motsvarande Ubuntu val **Failsafe**.

[3.1.1](#)

OBS! Du måste hålla musmarkören över terminalen, annars har den inte fokus (accepterar inte tangenttryckningar).

5.1.2 Tunna klienter (SunRay-terminaler)

De kvarvarande Solarissystem som finns i datorsalarna, är inte arbetsstationer, utan tunna klienter. Alltså, det du ser på bänken är inte en hel dator, utan bara en enhet med tangentbord, mus och bildskärm. Alla program körs på en central server, som användarna delar på. Detta har många fördelar – exempelvis är klienterna helt ljudlösa – men det är en del saker man måste tänka på

Du måste logga ut. Det räcker inte att stänga av klienten för att logga ut. När man sätter på den igen, så kommer du fortfarande att vara inloggad, och alla program fortsätter köra.

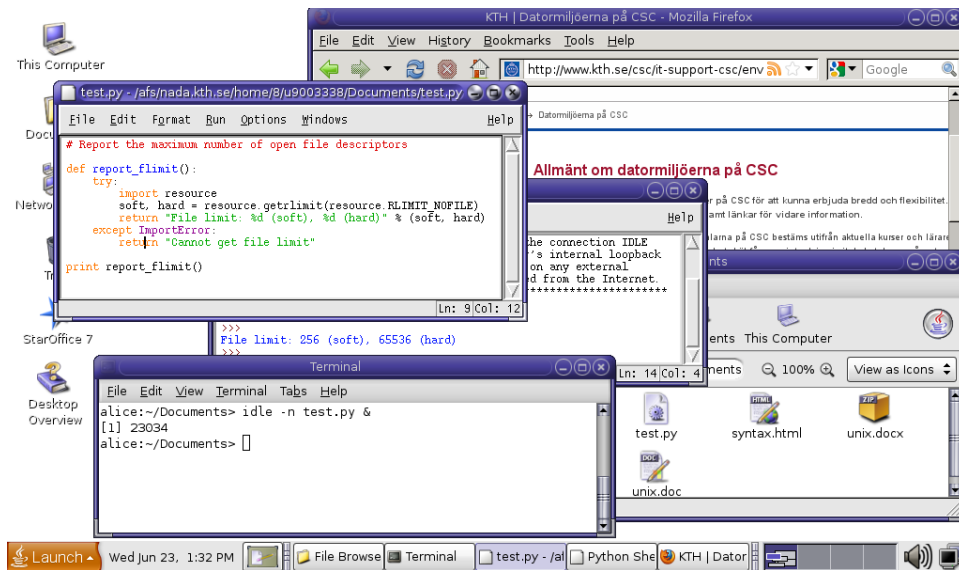
Ifall man har problem att logga ut, så kan man använda kortkommandot `Ctrl-Alt-backsteg` `backsteg` (håll nere `Ctrl` och `Alt`, tryck två ggr på `backsteg`). Detta dödar din inloggning. Alla program dödas också, och **du kan förlora öppna filer, om de inte sparats!**

Alla delar på resurser. Ifall du startar program som kräver mycket minne, eller andra resurser, så påverkas de andra användarna.

Enstaka program kan behöva köras på speciellt sätt. Exempelvis behöver man köra IDLE (för att [programmera Python](#)) som `idle -n` ifall flera användare kör programmet samtidigt. 9.2.1

5.1.3 Java Desktop System (JDS)

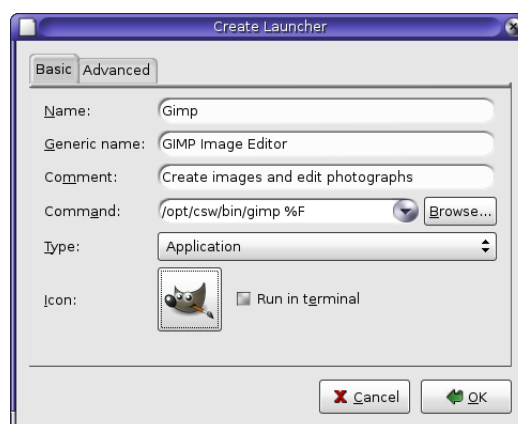
I Java Desktop har man endast en panel, längst nere i fönstret (om man inte flyttar den), där applikationer och inställningar ligger under en meny istället för bredvid applikationerna. (*Launch*-knappen, fungerar ungefär som MS Windows *Start*-knapp).



7.1.8

CSCs Solarissystem har utvecklats under många år. Eftersom man vill att flera olika versioner av program ska kunna köras i miljön, så bygger det till en ännu större del än de andra på [moduler](#). Detta gör att man bortsett från de mest grundläggande programmen, måste se till att moduler för de program man vill använda laddas när man loggar in, och att man själv måste lägga in de *launchers* som behövs för att starta dem¹

Exempelvis, för att få en genväg till programmet GIMP på skrivbordet, så kan du högerklicka och välja **Create Launcher**.



För att veta vilka värden du ska fylla i, så behöver du viss information. Programmet man ska köra heter `gimp`, så i en terminal skriver du

¹Alternativt, så kan man givetvis starta ett terminalfönster, och där lägga till moduler och sedan starta programmet.

5.1 Inloggning

```
> which gimp
no gimp in /usr/local/bin /usr/local/bin/X11 [...]
```

Tyvärr hittades inte `gimp`, eftersom programmet ligger i en modul. Alltså gör du

```
> whichmodule gimp
Module commands which gives access to different
versions/variants of the command "gimp":
  module add csw      (= module add csw/SunOS-5.10)
                       Alternative: csw/1.2
                               csw/1.3
  module add hacks    (= module add hacks/0.0)
  module add sfw      (= module add sfw/5.10)

> module add csw

> which gimp
/opt/csw/bin/gimp
```

Sedan kan du fylla i de värden som behövs.²

5.1.4 Flyttbara media (USB-minnen etc)

Eftersom många tunna klienter i sal kör på samma dator (som står i serverhallen), så blir hanteringen av USB-minnen tyvärr lite extra krånglig. Ett minne monteras i en underkatalog till `/tmp/SUNWut/mnt/` som heter samma sak som ditt användarnamn. I en underkatalog till den, hittar du de enheter du har monterade.

För att ta bort minnet, så använder man kommandot³

```
> /opt/SUNWut/bin/utumount \
-u /tmp/SUNWut/mnt/«användarnamn»/«katalog»
```

OBS! Filnamn med icke-engelska tecken, som t ex Å, Ä, Ö, kan ge problem. Det bästa är att begränsa sig till "vanliga" (dvs engelska) tecken.


²Ikonen finns i `/opt/csw/share/icons/hicolor/scalable/apps/gimp.svg`.

³Om man skriver allt på en rad behöver man inte tecknet \

6 Kommandoraden i UNIX

I ett terminalfönster kan man utföra saker som att flytta filer, skapa eller ta bort kataloger, mm. Detta görs genom att man skriver kommandon, som datorn sedan utför. För den som aldrig tidigare arbetat på det sättet, kan det kännas primitivt att skriva kommandon för att se vilka filer som finns i en katalog, och med dagens UNIX-system går i stort sett alla såna "enkla" saker att göra med exempelvis filhanterare. Det finns dock fördelar med ett terminalfönster (se nedan), och att lära sig använda det är att rekommendera för alla som läser någon form av programmeringskurs, vilket du som läsare rimligen gör.

6.2

Ett av de första kommandon du använder bör vara `kpasswd`  för att byta lösenord. Övriga kommandon går igenom i avsnitt 6.3, som kan användas som en referens.

Några fördelar med en kommandorad:

- Det går snabbt att arbeta. Även om inlärningströskeln är hög, så går det väldigt snabbt att utföra uppgifter, så länge man vet hur man uttrycker det man vill göra på kommandoraden.

Jämför med att använda ett kortkommando såsom `Ctrl-O`, istället för att sträcka sig efter musen och navigera sig till menyvalet **File** → **Open**.

- Det fungerar väl över nätverk, när man är fjärrinloggad.
- Det är ett effektivt sätt att ge instruktioner när man ska berätta vad någon ska göra. Även om samma saker utförs, så är det lättare att säga "Skriv `rm -r ~/.mozilla/` i ett terminalfönster", än att säga


Öppna en filhanterare. Navigera till hemkatalogen genom att klicka på verktygsradens *Home*-knapp. Visa dolda filer genom att välja *View* → *Show Hidden Files*. Markera katalogen *.mozilla*. Högerklicka och välj *Delete*.

- Det går att göra egna småprogram, *script*. Ett script är en sekvens av kommandon, som ska utföras precis som om man skrev in dem i ett kommandofönster¹.

Eftersom det går att anropa ett script precis som ett inbyggt kommando, så gör man ingen skillnad mellan "användarfiler" och "inbyggda program". Detta gör att man på ett förhållandevis lätt sätt kan utvidga ett UNIX-system med egna funktioner. Det är också en anledning till att UNIX är populärt bland programmerare.

När man arbetar med kommandoraden, så behöver man tänka på ett sätt som passar för att utföras i olika steg. Det man utför är oftast samma saker som man gör i ett grafiskt program, men man gör dem (delvis) med olika program. Antag att man vill skriva ut några filer. Man måste alltså:


1. Gå till katalogen där filerna finns. (*Program: cd*)
2. Lista alla filer. (*Program: ls*)
3. Skicka de filer man vill skriva ut till skrivaren. (*Program: olika*²)

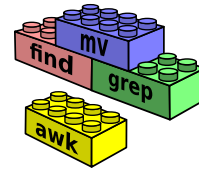
¹Ett *script*  [7.1.3] är i princip ett mindre program. Skillnad mellan vad som anses vara ett script eller ett program är flytande, och beror snarare på hur komplicerat det hela är, än någon teknisk skillnad mellan de två.

²Detta görs naturligtvis på olika sätt, beroende på vilken typ av fil det är.

6.1 Grundläggande: skal, prompt, sökväg

Att lära sig arbeta med kommandoraden består dels i att man lär sig vilka några av de vanligaste programmen är, dels i att man lär sig hur man sätter ihop de olika programmen för att bygga upp det man vill göra.

Ett av de allra viktigaste verktygen för att "sätta ihop" program är [omdirigering](#) .



[7.1.5](#)


6.1 Grundläggande: skal, prompt, sökväg

Det finns flera olika *skal*, alltså program som tolkar det man skriver i terminalfönstret. På CSC använder alla nya konton *bash*, men äldre konton använde *tsh*. De enskilda program vi går igenom i detta kapitel fungerar likadant i bägge, men för det vi gör i nästa kapitel kan det spela roll vilket som används.

Du kan se vilket skal du använder (eller byta) med kommandot `chsh` (**ch**ange **sh**ell).


Det första man ser när man öppnar ett terminalfönster är ledtexten (eng. *prompt*). Ledtexten ser i allmänhet ut som

```
användare@dator:sökväg>
```

där *dator* är namnet på den dator man använder, och *sökväg* är den katalog man står i (mer om det nedan). Utseendet på ledtexten kan ändras; ifall det står något annat använder man inte CSCs default-inställningar, se [hur man återställer dem](#) .

[10.5.4](#)

För att spara plats, så används här tecknet `>` som ledtext, i de fall då det inte är nödvändigt att veta dator och sökväg. I något fall kan vi även använda `användare:sökväg>`, för att göra det tydligare vilken användare som gör vad.

Det finns många verktyg inbyggda i skalprogrammet som ska göra det lättare att arbeta med det. Ett av de viktigare är [tab completion](#)  (på svenska *filnamnsexpansion*, men termen används sällan).

[7.1.4](#)

6.2 Byta lösenord (kpasswd)

Det första kommando du bör använda, om du inte redan gjort det, är `kpasswd`,³ för att byta ut det (oftast framslumpade) lösenord du fått till ett du kan minnas. Programmet frågar först efter det gamla lösenordet, sedan två gånger efter det nya. *Du ser ingen indikation på vad du skriver in*. Varken de tecken du skriver, eller någon annan symbol, kommer att visas.

`kpasswd` förkastar (med ett felmeddelande) lösenord som är alltför lätta. Det bästa är att ha några stora tecken, några små, någon siffra eller tecken som `&/()+!-,.` osv. Inga blanka tecken dock! Ej heller tecken som `ÅÄÖ`, eller andra tecken som inte ingår i ASCII!



Observera att de flesta på KTH har flera olika datorkonton. Man måste komma ihåg att ifall man byter lösenord i ett system, så byts det inte automatiskt i andra.

³På Solaris användes tidigare `passwd`, som också gav råd om hur man väljer ett bra lösen. `passwd` fungerar fortfarande, men endast på Solaris-systemen.


6.3 Kommandon


Kommandonamn i UNIX är ofta korta, eftersom man ska kunna skriva dem snabbt. Skulle man namnge dem idag, så skulle man kanske välja att bygga systemet så att man kunde anropa dem både med långa och korta namn, dvs skriva antingen *mv* eller *move*. Men på den tiden (70- och 80-talet) så var inte bara datorerna långsammare, utan det var också innan man kunde "slösa" bort (dyrt) minne på att duplicera informationen, eller sådan lyx som exempelvis kommandohistorik.

Motivering (och kanske mnemonik) till några av de vanligare kommandona⁴ finns i [tabell 1](#).


Kommando	Mnemonik	Användning
cd	change directory	ändra aktuell katalog
pwd	print working directory	visa aktuell sökväg
mkdir	make directory	skapa katalog
rmdir	remove directory	ta bort (tom) katalog
mv	move	flytta fil/katalog
cp	copy	kopiera fil/katalog
rm	remove	ta bort fil, med <code>-r</code> även (icke tom) katalog
cat	concatenate	visa filer som text
more	-	visa filer (som text), en sida i taget
less	-	<i>less is more</i> – dito, men med mer finesser
ln [-s]	link	skapa (symbolisk) länk
fs	file system	AFS-kommandon och -rättigheter
chmod	change mode bits	UNIX-rättigheter
man	manual	dokumentation, se (sid 63)  10.1
apropos	-	söka program efter nyckelord, (sid 64)  10.1

Tabell 1: Kommandoöversikt

[10.1](#) Mer utförlig dokumentation än vad som går igenom nedan, finns för samtliga kommandon i form av [man-sidor](#)  (manualsidor visas med kommandot *man*, därav namnet).

[6.4.1](#) Du kan styra beteendet för de flesta kommandon genom att skicka med en eller flera [väljare](#) . Exempel: `ls -a` visar även dolda filer.

6.3.1 Förflyttning i filträdet (cd och pwd)

[6.4.2](#) För mer information om hur kataloger organiseras, se avsnittet om [filträdet](#) .

För att inte alltid behöva använda den *absoluta* sökvägen (sökvägen ända från roten), så brukar man använda *relativa* sökvägar, som utgår ifrån den katalog man just nu "befinner sig i". Står man i sin hemkatalog `~`, så är det aktuell katalog, och `Private/README` betyder alltså filen `README` i underkatalogen `Private`, dvs `~/Private/README`.

För att byta katalog använder man kommandot `cd`:

⁴Motivering till några av de mer avancerade kommandon-namnen finns på ett Usenet-FAQ för UNIX: *What does {some strange unix command name} stand for?* (<http://www.faqs.org/faqs/unix-faq/faq/part1/section-3.html>)

Usenet, eller nyhetsgrupper (*newsgroups*), är ett textbaserat diskussionssystem.

6.3 Kommandon

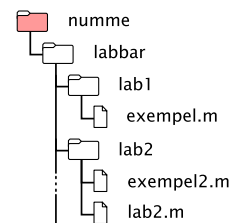
```
dator:~> cd Private
dator:~/Private> cat README
[...]
dator:~/Private> cd /usr/local/
dator:/usr/local> cd ..
dator:/usr>
```

För att det ska vara lättare att se var man befinner sig, så skrivs det även ut i ledtexten. För att se den absoluta sökvägen (istället för den relativt hemkatalogen), så kan man använda kommandot `pwd`.

På samma sätt som man kan använda `..` (punkt-punkt) för att nå ett steg upp i hierarkin, så kan man använda `.` (punkt) för aktuell katalog. Normalt sett behöver man inte det – det finns ingen anledning att skriva `cd ./Private` då `cd Private` betyder samma sak. Men ifall man exempelvis ska kopiera filer till aktuell katalog, så använder man `.` för att berätta vart de ska kopieras.

Man kan även gå flera steg på en gång. Antag att man befinner sig i katalogen `numme` i filträdet till höger. Då kan man använda följande kommandon för att först gå direkt till katalogen `lab1`, sedan upp ett steg, och ner i `lab2`.

```
dator:~/numme> cd labbar/lab1
dator:~/numme/labbar/lab1> cat exempel.m
[...]
dator:~/numme/labbar/lab1> cd ../lab2/
dator:~/numme/labbar/lab2>
```



För att slippa skriva in långa sökvägar, använder man lämpligen [tab completion](#) ⁵.

[7.1.4](#)

6.3.2 Lista filer (`ls`)

För att se vilka filer en katalog⁶ innehåller använder man

```
> ls [katalog]
```

För att kunna se vad som är en fil, en katalog, etc, så kan man ange väljaren `-F`, eller `--color`. I det första fallet, så läggs ett tecken till efter filnamnet, i det andra visas texten med annan färg. De vanligaste typerna är

Tecken	Färg	Betydelse
*	grön	program (exekverbar fil)
/	blå	katalog
@	cyan	(symbolisk) länk

För att hjälpa dig att snabbt se vad som är en fil eller katalog, så ställer de flesta system idag automatiskt in ett [alias](#) ⁵, som automatiskt lägger till någon av dessa väljare när man bara skriver `ls`.

[7.1.6](#)

Det finns många inställningar vars syfte är att hjälpa dig slippa skriva onödiga detaljer, eller avbryta ett annat kommando för att leta rätt på ett filnamn. Ett «tab»«tab» ([tab completion](#) ⁵) används ofta istället för `ls`.

[7.1.4](#)

Du kan även använda [jokertecken](#) ⁵ (eng *wildcards*) för att lista "allt som börjar på 'a'", eller (mer komplicerat) "allt i katalogen 'labbar' som börjar med 'lab' och slutar med '.m'":

[7.1.2](#)


⁵På svenska "filnamnsexpansion", men engelska termen används oftare.

⁶Ifall man inte anger en katalog antas aktuell katalog.

```
> ls a*
> ls labbar/lab*.m
```

Filer som börjar med "." (punkt) är *dolda*, dvs de visas vare sig när man listar filer i terminalfönstret, eller i ex. filhanterare. För att se även dolda filer, använd väljaren `-a`.

För att se storlek, datum och mer information om filerna, använd `ls -l`.


Det finns många, många andra väljare. `ls --help` visar två sidor(!) med de vanligaste. För ännu mer information (och detaljer som inte framgår av `--help`), se [man-sidan](#) .

[10.1](#)

6.3.3 Skapa och ta bort kataloger (`mkdir` och `rmdir`)

För att skapa en katalog används kommandot `mkdir`. Som vanligt kan man använda långa sökvägar, om man så önskar


```
> mkdir ny_katalog
> mkdir /lång/sökväg/till/ny_katalog
```

För att ta bort en katalog används `rmdir`, men *katalogen måste vara tom*. Även dolda filer (filer vars namn börjar med ".") måste vara borttagna. Det är därför vanligt att man istället använder kommandot `rm`  för att rekursivt ta bort kataloger.

[6.3.5](#)

6.3.4 Flytta och kopiera filer (`mv` och `cp`)

För att flytta en fil, eller byta namn på den (vilket anses vara ett specialfall av att flytta den), så använder man `mv`. Det är lätt hänt att råka skriva över en fil, det kan vara en bra vana att använda `-i`, då krävs det att man bekräftar innan det görs.

Kopior skapas med `cp`. Det går att kopiera kataloger inklusive innehåll (rekursivt) med `-r`, men det är lätt att göra ett misstag. Vad betyder det att kopiera en [länk](#) ? Får man en ny länk, eller en kopia av målet? Andra kommandon (`tar`, `rsync`) gör rekursiv kopiering säkrare. Se dokumentationen, så du inte gör fel. Även `cp` har en `-i`-flagga.

[6.3.8](#)

6.3.5 Ta bort filer (`rm`)

En fil raderas med `rm`. Med väljaren `-r` kan man även radera hela kataloger *rekursivt*, dvs först ta bort deras innehåll⁷, och sedan ta bort katalogen.

Varning! Det kan vara **mycket farligt** att använda `-r`. Det klassiska misstaget är att skriva exempelvis

```
> rm -r . thumbnails/
```

Varning

när man vill ta bort katalogen `.thumbnails`. Eftersom man råkade skriva ett mellanslag efter punkten, så tas först *allt i aktuell katalog bort* – man har ju skrivit `rm -r ..` Sedan får man ett felmeddelande eftersom det inte finns någon katalog `thumbnails/` att ta bort.

⁷Rekursivt", eftersom ifall det finns underkataloger måste dessa tas bort, vilket betyder att deras underkataloger måste tas bort...

6.3 Kommandon


6.3.6 Visa textfiler (`cat`, `more`, `less`)

Kommandot `cat` visar innehållet i filer. Namnet kommer av att ifall man anger flera filer, så *konkateneras* innehållet (slås ihop), och alla filer visas, den ena efter den andra.

Ifall innehållet inte ryms på en sida kan man istället använda kommandot `more`, som visar en sida i taget. Även om `more` har flera finesser såsom sökning (man trycker / och skriver det man söker efter), så kan man inte backa tillbaka texten. Därför använder många istället `less`⁸. Till skillnad från `more` måste man explicit avsluta `less` (med `q`), programmet avslutas inte när sista filen visats.

6.3.7 Utskrifter, utskriftskvoter

De flesta dokument skriver man ut grafiskt, från den applikation man använde för att skapa dem. Oftast går man via **File** → **Print...** För mer information om det, se respektive datormiljö.

Alla studenter har en utskriftskvot. Varje termin får du 400 sidor. För att skriva ut mer, så behöver du köpa extra sidor från [Systemgruppen](#) . Kontakta dem för information om hur.

[10.3.4](#)


Kvoterna återställs vid början av varje termin. Ifall du har köpt extra sidor, så kommer kvarvarande att falla över till nästa termin, men återstående "gratis" sidor gör inte det.

För att se hur många sidor du har kvar används kommandot `pkview`:

```
> pkview
User   Email                Limit by  Balance   Paid
alice  <alice@kth.se>      balance   212.00   400.00
```

För att skriva ut en fil med ren text används kommandot `a2ps`. Du kan behöva (beroende på skrivarinställningar) specificera pappersstorlek (A4), eftersom det amerikanska *letter* ofta är förvalt.

```
> a2ps -M A4 -1 filnamn
```

Du måste se upp så att filen har rätt [teckenkodning](#) . Annars kommer tecken som ÅÄÖ att bli fel. Ifall din fil använder UTF-8 kan du skriva ut den med⁹

[2.2](#)


```
> iconv -f utf-8 -t latin-1 < filnamn | a2ps -M A4 -1
```

Se man-sidan för `a2ps` ifall du vill veta vilka väljare som finns.

När du skickar en utskrift till skrivaren läggs den i en kö. Du kan se kön med kommandot

```
> lpq
```

⁸Kommandot `less` gjordes långt senare än `more`, och innehåller fler funktioner. Namnet kommer från talesättet *less is more*.

⁹Bakstreck (*backslash*) används i andra exemplet, för att kunna skriva varje program på ny rad. Det är valfritt att göra så. Omdirigering (tecknen `|` och `<`) går igenom i ett [senare avsnitt](#)  [7.1.5](#).

De flesta utskrifter gör du på den skrivare som står i salen. När du loggar in sätts den skrivare som står i eller närmast salen till *default*-skrivare (förvald skrivare). Både `a2ps` och `lpq` förstår väljaren `-P skrivare` för att använda en annan skrivare. De namnges på samma sätt som datorerna i salen. Exempel: *red* är skrivare i salen Röd, *musik* är skrivare i salen Musik.

Det går även att skicka godtycklig fil direkt till skrivaren, med `lpr`, men ifall du inte vet att det du skickar är en PostScript-fil¹⁰ kommer det förmodligen ut konstiga tecken, och **din skrivarkvot kan tömmas väldigt snabbt**.

För att ta bort felaktiga utskrifter, använder du kommandot

```
> lprm JobbID
```

där *JobbID* är ett nummer du får från `lpq`.

Utskrifter kan även hanteras grafiskt, se respektive miljö.

6.3.8 Symbolisk länk (`ln -s`)

En symbolisk länk¹¹ är en fil som "pekar på" en annan fil. Detta gör att man kan få tillgång till samma fil (eller katalog) på flera ställen i filsystemet utan att skapa kopior av den. Antag till exempel att du vill arbeta med en labbkamrat på samma programkod. Ifall du skapar en länk, `numme`, som pekar på kamratens katalog med koden, så kan ni bägge komma åt den utan att du behöver byta katalog till hans hemkatalog.

En viktig skillnad mellan symboliska länkar och MS Windows "genvägar" (*shortcuts*) är att program inte behöver skrivas så att de tar hänsyn till symboliska länkar. Detta gör att de fungerar i stort sett överallt – det är integrerat i systemet, snarare än i det enskilda programmet.

Till skillnad från de *alias* som finns i Mac OS X kan man inte flytta på målet för en fil.

Exempel: Antag att *alice* vill arbeta med *bob* i katalogen *numme*. Då kan hon skapa en länk till katalogen, och sedan komma åt filerna via den. Notera skillnaden mellan att lista länken (`ls numme`) och att lista innehållet i den katalog den pekar på (`ls numme/`).

```
> ln -s ~bob/numme numme
> ls -F numme
numme@
> ls numme/
numme/lab1.m
numme/lab3.m
> cat numme/lab1.m
[...]
> ls -l numme
[...] numme -> /afs/nada.kth.se/home/5/ulqbad95/numme
```

¹⁰PostScript (<http://en.wikipedia.org/wiki/PostScript>)

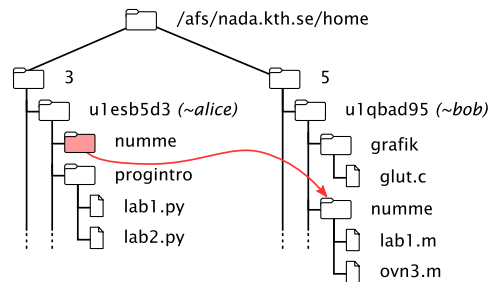
¹¹En *icke* symbolisk länk, eller "hård" länk (eng. *hard link*) skapas genom att man inte använder väljaren `-s`. En hård länk är den referens katalogen har för att associera ett namn med den faktiska fil som finns i filsystemet. En fil kan ha flera hårda länkar, och filen försvinner inte förrän den sista tas bort. (Symboliska länkar kan som jämförelse peka på filer som inte finns.)

Hårda länkar stöds inte av AFS, och används sällan på CSC. Då de även kan medföra vissa risker, så används ofta andra snarlika mekanismer, såsom *junction point* under NTFS, eller *alias* under Mac OS X.

6.3 Kommandon

Observera att *alice* inte har en kopia av *bobs* filer. Ifall hon ändrar på *lab1.m*, så ser även *bob* det, och ifall *bob* flyttar eller tar bort sin katalog så kommer inte längre *alice* åt filerna.

Att förstå alla detaljer i exemplet ovan tar ett tag att bena ut för den ovane. Exempelvis, så är det skalet som översätter *~bob* till dennes hemkatalog; "*~bob*" är inte en del av länken. Symboliska länkar kan även ge upphov till en del krångliga situationer. Antag t.ex. att *bob* skapar en länk inne i katalogen *numme*, som använder *..* (förälderkatalogen). Vad händer när *alice* använder den länken? Kommer hon åt sin hemkatalog, eller *bobs*? Vilket av följande är rätt? Båda?



```
> cd numme/  
> ln -s ../grafik bobs-kurs  
> ln -s ../progintro alices-kurs
```

När man förstår varför det ena fungerar, men inte det andra, så har man lärt sig en stor del av hur filhantering fungerar under UNIX.

Fällor:

- Ifall man kopierar en länk, så får man en kopia av målfilen. Väljaren `-d` på Ubuntu `cp` undviker detta, men denna väljare finns inte i t.ex. Solaris `cp` (om man inte även där använder GNUs `cp`; bägge finns installerade). Bättre är att använda `tar` eller `rsync`.¹²
- Om man skriver över länken (alltså, kopierar eller flyttar en fil till länkens namn), så skriver man över målet. Även här kan väljare ändra det, men det är bäst att tänka efter ordentligt!

6.3.9 Filrättigheter (fs, chmod)

På CSC ligger de flesta av dina filer under filsystemet **AFS** ¹³ där rättigheter fungerar lite annorlunda mot traditionella UNIX-rättigheter. De sätts på katalognivå (inte filnivå), med kommandot `fs setacl` (förkortas `fs sa`¹⁴), och visas med `fs listacl` (förkortas `fs la`).

7.2

```
fs sa katalog användare rättigheter  
fs la kataloger...
```

Rättigheterna syns i [tabell 2](#).

Exempel:

```
> fs la  
Access list for . is  
system:administrators rlidwka  
system:anyuser l  
alice rlidwka  
bob rl  
charlie rlidwk
```

¹²Bägge är komplicerade program, så ingetdera förklaras i detta häfte. Det är bättre att du [frågar om hjälp](#) ^{10.3} angående hur du använder dem.

¹³AFS, [Andrew File System](http://en.wikipedia.org/wiki/Andrew_File_System) (http://en.wikipedia.org/wiki/Andrew_File_System)

¹⁴ACL står för *access control list*, rättighetslista/åtkomstlista

	Betyder	Beskrivning
r	read	Läsa filerna i katalogen
l	list	Se vilka filer som finns i katalogen
i	insert	Skapa nya filer i katalogen
d	delete	Ta bort filer i katalogen
w	write	Ändra (innehållet i) filerna i katalogen
k	lock	Låsa filerna i katalogen
a	administer	Ändra i åtkomstlistan (ACLen) för katalogen

Tabell 2: AFS-rättigheter

Detta betyder att användaren *alice* har fulla rättigheter, liksom gruppen *system:administrators*¹⁵. *bob* kan läsa alla filer, men inte ändra dem. *charlie* kan läsa, ändra och ta bort filer, men inte ändra rättigheterna. Alla användare (*system:anyuser*) kan se vilka filer som finns, men inte läsa innehållet i dem.

```
> fs sa ~/kurser/numme bob rldw
> fs sa ~/kurser/numme charlie none
```

Detta ger *bob* rättigheter att ändra filerna i katalogen, och tar bort alla *charlie*:s rättigheter. Förutom bokstäverna kan man använda orden *read* (ekvivalent med *rl*), *write* (*rlidwk*, allt utom *a*), *all* (*rlidwka*) eller *none* (ta bort användaren från ACLen).

Skapar man en katalog, så kopieras ACLen från katalogen den ligger i. Ändrar man i en ACL påverkas däremot inte redan existerande underkataloger. Ska man ge rättigheter till exempelvis en labbpartner, så behöver man alltså först sätta rättigheter, sedan skapa underkataloger.

Du kan även behöva ändra på de traditionella UNIX-rättigheterna.

UNIX-filsystem (UFS, UNIX *File System*) har sedan länge haft rättigheter för filens (eller katalogens) ägare (*user*), den tillhörande gruppen (*group*) och alla andra (*other*). För varje kategori anges ifall man har rätt att läsa, skriva eller exekvera filen (allt öppna katalogen).

Dessa rättigheter visas med `ls -l`, eller i filhanteraren:

```
> ls -l anteckningar.txt
-rwxr-xr-x 1 alice usr 187 Apr 3 14:37 anteckningar.txt
```

Rättigheterna `-rwxr-xr-x` delas¹⁶ upp i grupper om tre:

user	group	other
rwx	r-x	r-x

där de enskilda bokstäverna betyder

Betyder	Ger rätt att
r read	läsa filen
w write	skriva i filen
x execute	exekvera (utföra) filen som program

¹⁵gruppen *system:administrators* innehåller administratörerna av AFS-volymens lokala cell. Alltså ges inte administratörerna av cellen *nada.kth.se* rättigheter i cellen *athena.mit.edu*. Egna grupper skapas med kommandot `pts`.

¹⁶Med undantag för det första strecket, som markerar något annat

6.3 Kommandon

Det lättaste sättet att lägga till respektive ta bort rättigheter för en fil är att skriva

```
> chmod +rättigheter filnamn
> chmod -rättigheter filnamn
```

men utförligare är kommandot

```
> chmod [referens][operator][mode] filnamn...
```


där *referens* bestämmer användare/grupp/andra, *operator* sätter eller tar bort rättigheter, och *mode* bestämmer läs/skriv/exekvering¹⁷.


Det är också vanligt att man (på samma sätt som när man programmerar i C) anger rättigheterna med tre eller fyra oktala siffror.¹⁸ Exempelvis kan man jämfört med ovan ta bort exekverings-flaggan genom:

```
> chmod 0644 anteckningar.txt
> ls -la anteckningar.txt
-rw-r--r-- 1 alice usr 187 2010-04-03 14:37 anteckningar.txt
```



UFS-rättigheter används fortfarande på lokala filer, dvs de som sparas på hårddisken i den dator man är inloggad på. Dessutom använder AFS *user*-delen av rättigheterna. Detta betyder att

För att	krävs AFS-rättighet	och filrättighet
skriva	skriva (w)	skriva (w)
ta bort	ta bort (d)	skriva (w)
exekvera	läsa (r)	exekvera (x)
läsa	läsa (r)	läsa (r)

Varning: Förutom att ha satt korrekta rättigheter, måste du för att kunna läsa filer ha autentiserat dig (identifierat dig för systemet). Detta görs normalt när man loggar in; då får man *Kerberos-biljetter* som varar i 10 timmar. Är man inloggad längre än så, kan man behöva använda kommandot `kauth` för att skaffa nya [biljetter](#) . Biljetter visas med `klist`. [7.2.1](#)

Man måste dessutom se upp så att katalogen inte är full. Se avsnittet om [kvoter](#) . [7.2.2](#)

6.3.10 Skriv text (echo)

För att skriva ut en textrad i terminalen, så används kommandot `echo`, eller under vissa omständigheter `cat`. Detta är främst användbart när man använder [omdirigering](#)  eller skriver [script](#) . [7.1.5](#)

[7.1.3](#)

6.3.11 Program och processer (kill, fg, bg)

När man startar ett program i terminalen exekveras (utförs) de instruktioner som finns lagrade i den fil som programmet finns sparad i. Dessa instruktioner körs i en eller flera processer.

Om du startar ett program som

¹⁷För utförligare förklaring, och fler exempel, se [Wikipedias sida om chmod](http://en.wikipedia.org/wiki/Chmod) (<http://en.wikipedia.org/wiki/Chmod>)

¹⁸To understand how this works, see the Wikipedia page for [file system permissions](http://en.wikipedia.org/wiki/File_system_permissions) (http://en.wikipedia.org/wiki/File_system_permissions#Octal_notation)

```
> firefox
```

så märker du att ledtexten försvinner tills programmet är klart. Detta kallas att programmet kör i *förgrunden* (eng *foreground*). Ifall programmet öppnar ett nytt fönster som du ska använda, kan detta vara onödigt. Då kan du istället köra programmet i *bakgrunden* (*background*):

```
> firefox &
```

Ifall man av misstag startat ett program som förgrundsprocess, och vill byta till att ha det i bakgrunden, så kan man sätta det i pausläge¹⁹ med `Ctrl-z`. Då visas meddelandet "Stopped" och programmet stannar. Om du då skriver `bg` körs det igång igen i bakgrunden.

Du kan även avsluta en förgrundsprocess, med `Ctrl-C`. Det kan dock vara farligt. Programmet får ingen chans att spara filer, städa undan tillfälliga data, etc. Det bör därför endast användas när det inte finns något annat sätt att avsluta programmet.

En process som körs i bakgrunden kan plockas fram igen, med `fg`. Både `bg` och `fg` kan ta ett argument, `%n`, där `n` är ett nummer som specificerar programmet (fås med `jobs`, och visas även när man trycker `Ctrl-z`).

Samma argument kan ges till `kill`, som avslutar en process på ungefär samma sätt som `Ctrl-C` (och är lika farligt).

Processer som startats i andra terminaler, eller på annat sätt, visas inte av `jobs`. Men man kan få information om dem med kommandot `top`. Det visar de processer som körs, normalt ordnat efter hur mycket processortid de använder för tillfället. Kolumnen `PID` visar processens id-nummer. Det kan användas som argument till `kill` (utan `%`).

Det finns även ett program, `xkill`, som man kan köra för att få klicka på ett fönster som tillhör ett program som skall "dödas".

Det finns mycket mer att skriva om processer, och hantering av dessa, men jag får hänvisa till manualsidorna för `kill`, `top` och `ps` (ett annat program för att visa aktuella processer).

6.4 Begrepp

6.4.1 Kommando, argument, väljare

Ett kommando består av *kommandonamnet*, följt av ett eller flera *argument*. En speciell typ av argument är en *väljare*. Något luddigt, så berättar kommandonamnet *vad* som ska göras, väljaren styr *hur*, och argumenten *på vad*, *var* e dyl.

Mellan argumenten ska det vara ett eller flera mellanslag. Ifall mellanslag ska ingå i ett argument måste man *citera* det, dvs omge det med enkla eller dubbla citattecken, alternativt använda bakstreck (eng *backslash*) framför²⁰

```
> kommando "ett argument"
> kommando 'ett argument'
> kommando ett\ argument
```

¹⁹Kallas också att "suspend:a", från engelskans *suspend*.

²⁰Att använda " eller ' är en smaksak, men det finns tillfällen då det är skillnad mellan enkla och dubbla citattecken. Notera att ett bakstreck (*backslash*), "\", inte är samma tecken som ett vanligt snedstreck (*slash*) "/".

6.4 Begrepp


Väljare (eng. *[command line] option; parameter; flag*) är oftast en bokstav med inledande "-", eller ett ord med inledande "--". Vissa kommandon accepterar att man slår ihop väljare (-sF istället för -s -F). Oftast kommer väljarna före övriga argument, men inte alltid.

Det händer också att väljare kräver ett *värde*. Exempel:

```
> find Private/ -type d
```

Kommandot *find* får väljaren *type* med argumentet *d* (för *directory*). *Private/* är här ett argument av övrig typ. Värt att notera är också att *find* alltså är undantag både till att väljare som utgörs av hela ord inleds med två streck, och till att "övriga argument" kommer efter väljarna.

Både kommandon och argument skiljer på gemener och versaler (små och stora bokstäver). *FIND* är inte alls samma sak som *find*, och det finns ingen väljare som heter *-Type*.

För att veta vilka väljare/argument som finns, och hur man anger dem, så kan man oftast anropa programmet med *-h*, *--help* eller liknande. Det finns även mer utförlig dokumentation, oftast i form av [man-sidor](#) .

10.1

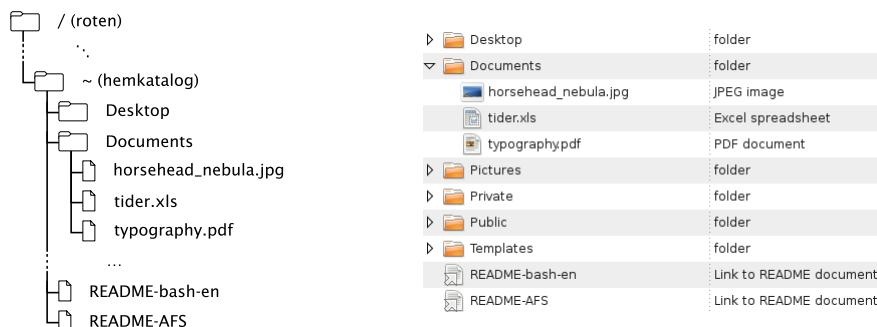
De flesta kommandon är tysta ifall de lyckas. Alltså, ifall kommandot misslyckas med att exempelvis skapa en katalog, så får man ett felmeddelande (mer eller mindre kryptiskt), men går allt bra så visas ingenting, utan man återvänder till ledtexten.

"No news, is good news"

6.4.2 Filträdet

All information i en dator lagras i *filer*. En fil består av binära data, och kan vara exempelvis ett program, eller ett textdokument. Alla tecken förutom "/" är tillåtna i ett filnamn, men använder man tecken som radbrytning eller * kan det vara jobbigt att skriva namnet. Filsystemet bryr sig inte om vad filen innehåller – text, bild, ljud, etc – det får de olika programmen hantera.

Filerna ordnas i *kataloger* (kallas även *mappar*), för att öka överskådligheten. Genom att placera kataloger i kataloger, bygger man upp en trädstruktur. De flesta filhanterare kan visa en sådan trädstruktur:



Den katalog som innehåller alla andra kataloger och filer heter *roten* (eng *root*), och betecknas med tecknet snedstreck, /. Övriga kataloger betecknas med namn, och separeras med snedstreck. För att beskriva var i filsystemet en fil (eller katalog) ligger, så måste man ange hur man tar sig från roten via underkataloger till filen. Beskrivningen kallas en *sökväg*, och ser ut till exempel så här:

```
/var/tmp/readme.txt
```

som betyder att filen `readme.txt` ligger i katalogen `tmp`, som ligger i katalogen `var`, som ligger i `/` (roten). För att visa denna fils innehåll i terminalen med kommandot `cat`²¹ skriver du:


```
> cat /var/tmp/readme.txt
```

Observera att även löstagbara enheter (CD-skivor, USB-minnen) stoppas in någonstans i filsystemet (vanligen någonstans under `/media/`). Det kallas för att enheten *monteras* (från eng. *mount*); normalt sker detta automatiskt.

Förutom den speciella katalogen `/` finns även ett par andra.

- ~ Din hemkatalog. Den katalog som innehåller dina filer och inställningar. Man kommer även åt andras hemkataloger med `~namn`, exempelvis `~alice/README-AFS` för att komma åt en fil i `alice:s` hemkatalog
- .
- .. Aktuell katalog, se nedan.
- .. Ett steg upp, "förälder-katalog" (*parent directory*)

Man kan givetvis lagra filer även på andra ställen än sin hemkatalog, men att ha en dedikerad hemkatalog hjälper dig att avgränsa det mesta av det du äger, från det andra äger eller som hör till systemet.

[7.2](#) På mindre UNIX-system ligger hemkataloger oftast under `/home`, exempelvis `/home/alice`. På CSC, med över 25 000 användare, så skulle det bli opraktiskt. Dessutom används [AFS](#) , så hemkataloger här ligger under exempelvis²²

```
/afs/nada.kth.se/home/3/ulak8b73/
```

Ofta kan man undvika att skriva de två första stegen, `/afs/nada.kth.se` och istället skriva `/home/3/ulak8b73/`.

Ifall man använder ett program för filöverföring, eller av annan orsak inte kan skriva `~namn`, så finns även en katalogstruktur som utgår från `namn`,

```
/afs/nada.kth.se/home-by-name/«x»/«y»/namn/
```


där `x` och `y` är de två första tecknen i användarnamnet, exempelvis `/afs/nada.kth.se/home-by-name/a/l/alice/`.


De vanligaste katalogerna visas i [tabell 3](#)

Skillnaden mellan `/bin`, `/usr/bin`, och `/usr/local/bin` (och motsvarande för `sbin` och `lib`) är huvudsakligen för att kunna administrera flera datorer. Man skiljer på tre olika nivåer, med ungefär samma kataloger. Program finns exempelvis i:

1. `/bin` som innehåller systemkritiska program; dessa behövs ifall man ska felsöka datorn i *single user mode* (felsäker uppstart).
2. `/usr/bin` som innehåller program som inte behövs för *single user mode*. Dessa är i princip likadana som motsvarande program på andra datorer med samma operativsystem. Det går att lägga denna katalog på en nätverksenhet, men detta görs inte på CSCs datorer. De flesta program finns här.
3. `/usr/local/bin` som är de program som eventuellt bara finns på denna dator.

[6.3.9](#)

För att sätta rättigheter, så används kommandona `chmod`  (under UFS,

²¹Kommandot "konkatenerar" (slår ihop, eng *concatenate(s)*) filer; `cat`  [6.3.6](#)

²²Den krångliga biten med `ulxxx` plockas från en central databas med KTH-id:n.

6.4 Begrepp

Katalog	Beskrivning
/	roten på filsystemet (nivå 1)
/usr	ej systemkritiska data, krävs inte för felsäker uppstart (nivå 2)
/usr/local	datorspecifika data (nivå 3)
/bin	systemkritiska program (även under nivå 2–3)
/sbin	systemkritiska administratörsprogram (även nivå 2–3)
/lib	programbibliotek för /bin
/dev	”filer” som representerar hårdvara
/etc	konfigurationsfiler
/home	hemkataloger
/media	tillfälliga media såsom CD-skivor, USB-minnen etc.
/tmp	tillfälliga filer (innehållet överlever inte omstart)
/var	variabla filer – sånt som förväntas ändras ofta (Loggar, låsfilen, köade utskrifter, etc)
/var/tmp	tillfälliga filer (städas med jämna mellanrum)
/opt	vissa tillägsprogram

Tabell 3: Katalogstruktur (urval)

UNIX *File System*), eller `fs sa` (under AFS). UFS hanterar rättigheter på [6.3.9](#) filnivå, AFS endast på katalognivå.

7 Fördjupning

7.1 Mer om kommandotolken

7.1.1 Kommandohistoria

Man vill helst slippa skriva om något, bara för att ändra ett tecken som blev fel. Kommandotolken kommer ihåg vilka kommandon man har givit, och man kan snabbt få tillbaka föregående med «pil upp».

Man kan visa hela historiken med `history`, och söka bakåt i den med `Ctrl-R`. Även andra kommandon finns tillgängliga, för att klippa, klistra och ändra det man skrivit. Det mesta är lånat från Emacs kortkommandon, och dokumentationen använder också dess terminologi. Därför ska man leta efter `C-r` istället för `Ctrl-R` i dokumentationen för att se detta, eller `M-u` istället för `Alt-u` för kommandot som gör om efterföljande ord till VERSALER.

7.1.2 Wildcards (jokertecken)

Många har tidigare sett `*.txt` som beteckning för "alla textfiler", dvs filer som slutar med ändelsen `.txt`. I UNIX kan man även använda `?` för att säga "ett tecken, vilket som helst"¹ och skapa grupper av tecken med `[]`. Gruppen `[aeiou]` skulle alltså matcha alla (engelska) vokaler², och `[a-z0-9]` alla (gemena) bokstäver och siffror³.

Exemplet nedan öppnar (endast) de två filerna `final.pdf` och `penguin.pdf` i *Document Viewer* (vars program heter `evince`).

```
> ls
README.txt      final.pdf
penguin.pdf    Tux-G2.pdf
> evince [a-z]*.pdf &
```

Värt att notera är att `*.*` inte är "alla filer". Det skulle matcha varken (1) `readme` (ingen punkt), (2) `.readme` (inget innan punkten) eller (3) `readme.` (inget efter punkten).

Ifall man vill använda ett specialtecken på kommandoraden som en vanlig bokstav, så behöver man skydda det:

```
> echo Vill du se en * se på...
Vill du se en Desktop Documents OldFiles Private
Public README-AFS se på...
> echo Vill du se en '*' se på...
Vill du se en * se på...
```

Man kan använda

<code>'argument'</code>	Skyddar alla tecken utom <code>'</code> .
<code>"argument"</code>	Skyddar alla tecken utom <code>"</code> och <code>\$</code> .
<code>\x</code>	Skyddar <code>x</code> mot behandling, där <code>x</code> är (nästan) vilket tecken som helst.

¹Så länge man inte menar tecknet `/` som måste skrivas explicit, eller `.` i början av ett filnamn.

²"y" är inte alltid en vokal i engelska.

³Strecket betyder "alla tecken mellan", men ordningen är inte helt självklar. Den beror på teckentabellen, se [ASCII](http://en.wikipedia.org/wiki/Character_encoding) (http://en.wikipedia.org/wiki/Character_encoding)

7.1 Mer om kommandotolken

7.1.3 Script

Ett script är ett program som inte kompileras från källkod, utan består av en textfil som tolkas rad för rad av ett program.⁴ Man kan känna igen dem genom att den första raden i filen börjar med tecknen `#!`, följt av sökvägen till det program som ska tolka skriptet. Sekvensen `#!` uttalas normalt "shebang". Ett minimalt script skulle kunna se ut så här:

```
#!/bin/sh
file /usr/bin/* | grep -i script
```

Mer avancerade script kan hantera argument, visa GUI:n, etc. De byggs upp med *if*-satser, *while*- och *for*-loopar och andra vanliga programmeringsverktyg.

Många script byggs upp med *pipelines* (se nedan), och det finns många små verktyg som kan användas för att flytta om ord eller liknande. Exempelvis kan man i sitt script stoppa in

```
... | tr -c -d 'A-Za-z ' | ...
```

för att ta bort allt utom (engelska) bokstäver och mellanslag.

7.1.4 Tab completion (filnamnsexpansion)


För att slippa skriva varje tecken i ett långt filnamn, eller för att bekräfta att man verkligen skrivit rätt på en sökväg, så kan man använda *tab completion*. Detta består i att kommandotolken fyller i så mycket av filnamnet som den kan. Om det t.ex. finns två filer som heter `readme.txt` och `read_this_first.txt`, så kan man skriva

```
> mv -i r<tab>m<tab>
```

Efter första `<tab>`-tryckningen, så kompletteras namnet så långt det är möjligt, dvs till `read`. Efter detta är det inte unikt, det kan följas av antingen `me...` eller `_this...`. När man skriver dit `m`, så är namnet unikt, och nästa `<tab>` fyller i hela namnet.

Ifall man inte skriver dit en bokstav till, utan trycker `<tab>` igen, så visas de kompletteringar som finns. Det fungerar även när man skrivit namnet på en katalog (inkl. efterföljande snedstreck). Då visas alla filer i katalogen. Att trycka `<tab><tab>` blir snabbt en vana på kommandoraden, för att se att datorn faktiskt kan se de filer man tror ska finnas där.

Tab completion förekommer även i andra program än kommandotolken, t.ex. har de flesta editorer någon form av det.

För den som använder *tcsh* fungerar det lite annorlunda. Där används inte `<tab>`, utan `?`. Se avsnittet om [gamla inställningar](#) .

[10.5.2](#)

7.1.5 Omdirigering

Om ett kommando skriver ut sitt resultat på skärmen, så kan man styra om det så att resultatet istället skrivs till en fil, eller skickas till ett annat program.

⁴Även om man kan prata om t ex MATLAB- och Python-program som script, så menar man vanligen de script som tolkas av samma program som man kör i terminalfönster, dvs *sh*, *bash*, *tcsh* etc.

För att skriva till en fil, använder man `>` eller `>>`. Det första skriver över filen, ifall den redan finns. Det andra fyller på i slutet av filen (kallas "append:ar", från engelskans *append*). Man kan även skicka in en fil till ett program, med `<`.

```
> ls > fil
> tr 'a-z' 'A-Z' < fil
```

Här sparas listan av filer på en fil, som sedan skickas till `tr` som översätter alla (engelska) bokstäver till versaler, och skriver resultatet på terminalen.

För att slippa använda en tillfällig fil, så kan man skicka data direkt mellan programmen, med en *pipe* (eller *pipeline*):

```
> ls | tr 'a-z' 'A-Z'
```

Termer

standard output Utdataström, den ström av data ett program skriver ut (till terminalfönstret, ifall inget dirigeras om). Kallas även "stdout".

standard input Indataström, varifrån programmet läser. Även "stdin".

standard error Felutskriftsström, används så att program kan visa fel även om stdout dirigerats om.

pipeline, pipe Det som binder ihop två program, och även tecknet `|`, som används för att göra det.⁵

Pipelines används för att bygga ihop mer komplicerade program. Ett par exempel:

```
> find . -name "*.doc" -print0 | xargs -0 gzip -9
> cat *.txt | \
sed -e 's/[^a-zA-Z ]/ /g' | \
tr 'A-Z' 'a-z\n' | \
grep '[a-z]' | \
sort -u | \
comm -23 - /usr/share/dict/words
```

Notera att bakstreck (eng *backslash*) har använts i andra exemplet, för att kunna sätta varje program på ny rad. Kortfattad förklaring:

1. Hitta alla Word-dokument, och komprimera dem
2. Skriv ut alla text-filer, filtrera bort icke-bokstäver, översätt till gemener med varje ord på egen rad, ta bort blankrader, sortera och ta bort dubletter, leta efter ord som inte finns i en ordlista. Puh!

Alltså, en rudimentär stavningskontroll. (Det finns bättre program för just det.)

⁵I andra sammanhang heter tecknet vertikalt streck eller *vertical bar*.

7.1 Mer om kommandotolken

Verktyg man bör känna till

sed Utför automatiskt textförändringar såsom sök-och-ersätt, o dyl. (Namnet kommer från *stream editor*.)

awk Mer avancerad textredigering. I princip arbetar *sed* rad för rad, men *awk* på flera rader. (Namnet från Aho, Weinberger, Kernighan, som gjorde programmet.)

grep Sök efter text med *reguljära uttryck*⁶. (Namnet från *global* / *regular expression* / *print*, namn på instruktioner i en äldre textredigerare.)

tr Översätt (*translate*) eller ta bort bokstäver. Byter bokstäver i en lista, mot bokstav på motsvarande plats i den andra (eller tar bort bokstaven, beroende på väljare).

diff Visar (rad för rad) skillnader mellan två textfiler.

7.1.6 Alias och funktioner

För att ge ett kort namn åt kommandon du använder ofta, kan du använda ett *alias*.

```
> alias ll='ls -laFrt'
```

Detta gör så att varje gång man skriver `ll`, så anropas `ls -laFrt` istället. Det är inga problem att ge argument till alias, exempelvis `ll Public/`.

Ifall man till sitt kommando behöver ge argument som inte kommer efter sista ordet, så kan man istället skriva en *bash*-funktion. Nedan kommer `$*` att ersättas med argumenten⁷

```
> lm () { ls -l $* | more; }
```

Normalt så skrivs alias och funktioner in i någon av *bash*s inställningsfiler, lämpligen `.bashrc`. Den `.bashrc` som skapas för nya konton på CSC innehåller en del exempel.

7.1.7 Exportera miljövariabler

Miljövariabler (*environment variables*) är ett sätt för olika program att berättas för varandra exempelvis vilken skrivare man ska använda, eller var man ska leta efter program. Systemet sätter vissa variabler när man loggar in, andra kan man sätta själv:

```
> export variabel=värde
```

(Ifall bara *bash* ska använda variabeln, så behövs inget `export`.)

⁶Reguljära uttryck (*regular expression*, *regex*) är ett mycket kraftfullt sätt att söka i text. Tänk på det som vad jokertecken vill bli när de blir vuxna. Det är ett verktyg som kräver mycket eftertanke – reguljära uttryck kan vara nästan oändligt komplicerade, och med de mer avancerade konstruktionerna är det mycket lätt att "skjuta sig i foten", dvs göra fel.

⁷Det kan dock vara värt att se efter vad "`$@`" är (inklusive citattecken), istället för att använda detta exempel.

7.1.8 Moduler

CSCs system har väldigt många program, och till skillnad från de flesta datorer i hemmet, så måste det⁸ även finnas flera olika versioner av program installerade. De vanligaste programmen finns tillgängliga i standardmiljön, men för att få tillgång till andra program kan man behöva använda så kallade *moduler*. Detta görs med kommandot

```
> module add modulnamn
```

där *modulnamn* är någon av de (många) moduler du får upp om du gör

```
> module avail
```

För att hitta vilken modul som ger tillgång till ett visst kommando, använder du

```
> whichmodule programnamn
```

Ifall man alltid vill ha tillgång till en viss modul så kan man använda

```
> module initadd modulnamn
```

eller manuellt lägga till *modulnamn* i filen `~/.modules`, som läses när du loggar in. `module help` visar fler modulkommandon.

7.2 AFS och Kerberos

Andrew File System är ett distribuerat nätverksfilsystem som man kan komma åt på ett säkert sätt från alla datorer som använder AFS. Versioner finns för alla vanliga operativsystem. Alla hemkataloger på CSCs UNIXsystem (inklusive Mac och Linux) ligger i AFS.

AFS organiseras i "celler" (eng *cell*), exempelvis *nada.kth.se*, *pd.c.kth.se* och *andrew.cmu.edu* (de motsvarar ungefär Internet-domäner). Celler innehåller "volym" (*volumes*); din hemkatalog är en volym. För att se vad volymen heter används `fs exa`:

```
> cd
> pwd
/afs/nada.kth.se/home/3/ulak8b73
> fs exa
File . (868903664.1.1) contained in volume 868903664
Volume status for vid = 868903664 named home.3.ulak8b73
Current disk quota is 200000
Current blocks used are 138294
The partition has 44513490 blocks available out of 746334433
```

Som synes ligger hemkatalogen i en volym vars namn matchar sökvägen, och heter `home.3.ulak8b73`.

Säkerhet sköts med systemet Kerberos. I det begär man en "biljett" (*ticket*), som ger åtkomst till en "tjänst" (*service*), exempelvis tillgång till AFS. Istället för att vara uppdelad i celler, så använder Kerberos termen "realm". Det är i praktiken väldigt liten skillnad mellan dem, bortsett från att realm:er

⁸Särskilt på Solaris-miljön, som utvecklats under flera årtionden

7.3 Rättigheter

skrivs med versaler. AFS-cellen `nada.kth.se` kräver alltså Kerberos-biljetter i realm:en `NADA.KTH.SE`.

Vanligtvis sker alla detaljer utan att användaren behöver göra något. När du loggar in får du "tokens" (en typ av biljetter, kallas vardagligt "AFS-plåtar") som ger dig rättigheter till din hemkatalog. Vanligtvis behöver man inte använda AFS eller Kerberos-programmen direkt särskilt ofta, men man behöver vara medveten om hur det fungerar, särskilt hur rättigheterna sätts. (Med `fs sa`, se avsnittet om [filrättigheter](#) ↪)

[6.3.9](#)

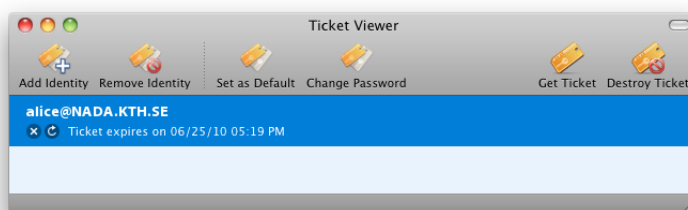
7.2.1 Biljetter

För att skaffa biljetter, använder du `kauth`⁹. Detta ger dig en TGT, *ticket-granting-ticket*. CSCs defaultinställningar är att din biljett är forwarderbar, och att du dessutom får AFS-tokens. Dessa kan annars skaffas med `afslog`¹⁰.

För att se dina biljetter använder du `klist`. På Ubuntu och Solaris kan du använda `klist -Tf` för att också se AFS-tokens; annars kan man använda `tokens` för det senare.

Den som vill komma åt AFS-volymer i andra celler, eller av annan anledning ha en biljett för en annan realm, kan se dokumentationen eller [fråga Systemgruppen](#) ↪ [10.3.4](#). Användbara kommandon är bl a: `kauth/kinit`, `klist`, `afslog/aklog`, `tokens`, `kpagsh`.

Det finns även grafiska program för att hantera biljetter, exempelvis *Ticket Viewer* på Mac OS X (ligger i `/System/Library/CoreServices`).



7.2.2 Kvot

Varje hemkatalog (eller annan AFS-volym) har en *kvot*, dvs en övre gräns för hur mycket utrymme man får använda. För att se hur stor kvoten är, använder man `fs listquota` (förkortas `fs lq`):

```
> fs lq katalognamn
```

Det spelar ingen roll hur många underkataloger som finns. Alla kataloger i en volym delar på utrymmet. Å andra sidan kan det finnas andra volymer monterade där, med egen kvot. Exempelvis, så är `~/OldFiles` en backup-kopia av ditt konto från gårdagen; dess kvota påverkar inte hemkatalogens.

7.3 Rättigheter

Det mesta om hur rättigheterna fungerar behandlades i avsnittet om [filrättigheter](#) ↪.

[6.3.9](#)

⁹På vissa system används `kinit` istället

¹⁰Eller `aklog`

Värt att nämna, är dock att att vissa tjänster på CSC (exempelvis webbserverar) ofta använder speciella "användare" för rättigheterna. Alltså, om man vill att endast webbservern ska ha rättigheter till en katalog så kan man sätta dem till användaren *wwwservice* istället för *system:anyuser*. **Detta ger inte fullgod säkerhet!** Eftersom alla användare som har en hemsida kan be webbservern läsa filerna, är detta möjligtvis att betrakta som *security by obscurity* (http://en.wikipedia.org/wiki/Security_by_obscurity).

7.3.1 Rättighetsgrupper

Kommandot `pts` används för att skapa egna rättighetsgrupper. Grupperna inleds med ägarens namn, på formatet «användare»:«grupp». Du kan alltså skapa en grupp för dina labbpartner, och sedan lägga till eller ta bort användare i gruppen, istället för att ändra på rättigheterna i många olika kataloger.

```
> pts creategroup ägare:grupp
```

skapar en grupp. Användare läggs till och tas bort med `pts adduser` eller `pts removeuser`, se filen `README-AFS` i din hemkatalog för mer information.

7.3.2 Säkerhetskopior – OldFiles

Alla hemkataloger (och de flesta andra volymer) säkerhetskopieras ("backas upp", från engelskans *backup*) dagligen.¹¹ Du kommer åt din senaste backup under katalogen `.OldFiles` i din hemkatalog. Det finns även en mjuk länk, utan punkten, så att man ska komma åt katalogen lättare, men ändå kunna säga exempelvis

```
> find * -name "brev.doc"
```

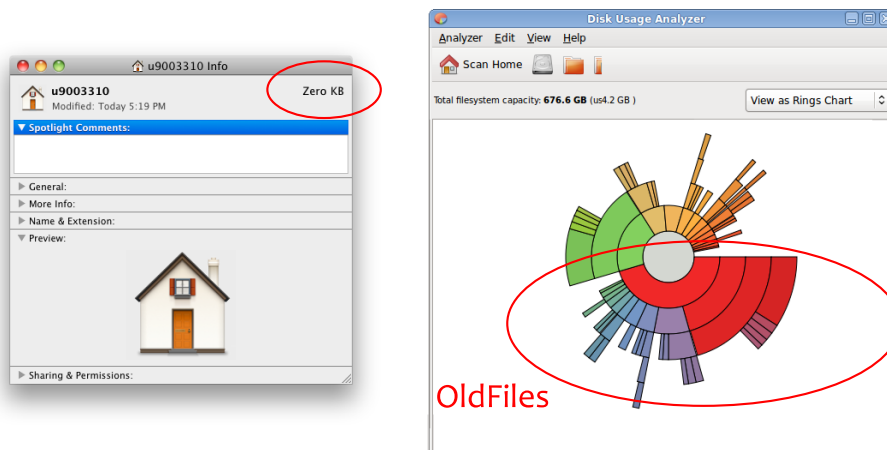
utan att `OldFiles` söks igenom.

Nackdelen med att ha det så, är att kommandon som inte förstår sig på AFS kan ge felaktiga (eller onödiga) resultat. Om man söker igenom hela sin hemkatalog (`find .` istället för `find * ovan`), så får man ofta två träffar – en onödig i `.OldFiles/samma/sökväg/som/andra/resultatet`. Ifall man ska kopiera sin hemkatalog, så måste man vara noggrann för att inte få med en extrakopia.

Även program som visar hur mycket utrymme man använt, visar ofta fel. *Finder* (Mac OS X) kan under vissa omständigheter tro att hemkatalogen inte tar någon plats alls, medan *Disk Usage Analyzer* (*Baobab*) tycker att `OldFiles` fyller halva kontot.

¹¹Den enda säkerhetskopior du kommer åt själv, är gårdagens, men CSC sparar kopior i ett rullande schema, för att kunna gå tillbaka varje dag i ett par veckor, varje vecka innan det i ett par månader, etc. Har man tur kan en förlorad fil gå att hitta, men man måste ange ett så begränsat tidsintervall som möjligt!

7.3 Rättigheter



Tekniskt, så finns backup för en AFS-volym i en särskild volym, «volym».backup. Exemplet ovan hade volymen `home.3.ulak8b73` – dess backup skulle alltså heta `home.3.ulak8b73.backup`, och OldFiles skulle (åter)skapas genom

```
> fs mkm ~/.OldFiles home.3.ulak8b73.backup
> ln -s .OldFiles ~/OldFiles
```

7.3.3 Delade kataloger

För att skapa en katalog där du och dina labbkompisar kan spara gemensamma filer, så skapa katalogen på något av era konton och sätt sedan upp rättigheterna så att alla kan komma åt den. Skapa sedan länkar till katalogen.

Även om katalogen kan ha vilket namn som helst, och ligga i någon underkatalog, så är det ofta bäst att kalla den samma sak som kursen, och lägga den i hemkatalogen.¹²

Notera att underkataloger till kurskatalogen ska skapas *efter* katalogen fått sina rättigheter. Annars ärver de inte rättigheterna.


Kommandot `course labdir` gör det lätt att skapa delade kataloger. Se exempel 2 nedan.

Exempel Användaren *alice* vill arbeta med sina kamrater *bob* och *charlie* i kursen "C-programmering". Hon skapar katalogen och sätter rättigheter (*write* och *rlidwk* betyder här samma sak).

```
alice::~> mkdir ~/cprog09
alice::~> fs sa ~/cprog09 bob rlidwk
alice::~> fs sa ~/cprog09 charlie write
```

Sedan får *bob* och *charlie* skapa mjuka länkar till katalogen

```
bob::~> ln -s ~alice/cprog09 ~/cprog09
charlie::~> ln -s ~alice/cprog09 ~/cprog09
```

Exempel 2 För att göra samma sak med `course labdir` , så kan nästan samma kommando användas av alla tre användare:

```
alice:~> course labdir cprog09 bob charlie
bob:~> course labdir cprog09 alice charlie
charlie:~> course labdir cprog09 bob alice
```

Man o, alla anger de andras användarnamn. (Katalogen läggs i hemkatalogen till den som använder kommandot först.)

Katalogen kommer att få precis samma egenskaper som i första exemplet, förutom att alla kommer att ha fulla accessrättigheter.

7.3.4 Olika rättigheter i samma katalog

Vissa filer i din hemkatalog behöver läsas när du loggar in, innan du autentiserat dig. Andra filer kan innehålla information som ska vara hemlig (lösenord, andras e-postadresser). Filrättigheter i AFS sätts på katalognivå, så vad gör man då?

Lösningen är att ge *list*-rättigheter (men inte läsrättigheter) till *system:anyuser*, och sedan skapa en allmänt läsbar underkatalog där man kan lägga icke-hemliga filer. För varje fil som ska vara allmänt läsbar, skapar man sedan en symbolisk länk som pekar på filen i underkatalogen. Då kan program komma åt de allmänna filerna genom länkar i den icke allmänt läsbara katalogen.

Ett exempel på hur man sätter upp detta:

```
> cd                gå till hemkatalogen
> mkdir Public      skapa underkatalog
> fs setacl Public system:anyuser read gör den läsbar
> mv .bashrc .bash_profile Public flytta filer dit
> ln -s Public/.bashrc skapa länkar
> ln -s Public/.bash_profile
```

10.5.4

Nya konton på CSC sätts upp på detta sätt, och du kan [återställa detta](#)  ifall det blivit förstört.

7.4 Hemsida (public_html)

För att skapa en hemsida lägger du en fil som heter `index.html` i katalogen `public_html`. Filen ska innehålla HTML-kod.¹³

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<html><head><title>Minimal</title></head>
<body>
<p>Detta &auml;r en minimal sida med HTML-kod.</p>
<p>En l&auml;nk: <a href="http://www.kth.se">KTH</a>.</p>
</body></html>
```

Hemsidan kan du nå på `http://www.csc.kth.se/~<användarnamn>/`.

Observera att `index.html` är speciell. Namnet `http://.../katalognamn/` tolkas¹⁴ som att man söker filen `index.html` i den katalogen. Andra filnamn behöver skrivas ut.

¹²Annars måste alla ha rättigheter att läsa alla kataloger mellan hemkatalogen och kurskatalogen – annars hittar de den inte.

¹³För att lära dig HTML kan du t ex läsa [w3schools](http://www.w3schools.com/) (http://www.w3schools.com/) guider (på engelska).

Det går även att se på koden till sidor på Internet. En del sidor (exempelvis KTHs) är dock automatiskt genererade, vilket inte alltid ger lättläst HTML-kod.

¹⁴Inställningarna kan variera, beroende på vilken av CSCs webbservrar som används.

7.5 Fjärrinloggning (ssh)

7.5 Fjärrinloggning (ssh)

För att komma åt CSCs system hemifrån, kan du använda någon form av fjärrinloggning. Det finns även verktyg för att flytta filer, antingen till eller från CSC. Givetvis fungerar det även att flytta filer med USB-minne, CD-skiva eller dylikt, men detta kräver att du är på plats, dvs kan sätta dig vid en CSC-dator. Detta är inte alltid möjligt, och det är inte heller nödvändigt.

För fjärrinloggning använder man någon form av SSH (*Secure SHell*), eller (för filöverföring) SFTP eller SCP (*SSH File Transfer Protocol, Secure Copy*)¹⁵

7.5.1 Terminalinloggning (SSH)

För den som använder Mac OS X, Linux eller någon UNIX-variant hemma, har förmodligen dessa program installerade. För Windows, så finns ett flertal gratisprogram, det vanligast är *PuTTY*¹⁶.

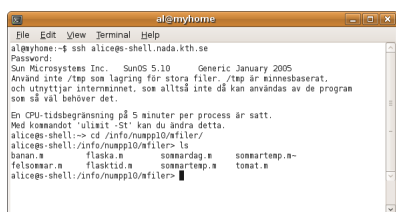
När man loggar in, så gör man det från *lokal dator* (klient) till en *värddator* (server). Även om det i princip skulle gå att göra mot vilken sals-dator som helst, så är detta inte tillåtet. Du kan istället använda någon av de datorer som är reserverade för ändamålet:

s-shell.csc.kth.se – en server som kör Solaris, CSCs traditionella datormiljö, och även det som körs på webbservrar, o dyl.

u-shell.csc.kth.se – en server som kör Ubuntu Linux, som numer körs i flertalet datorsalar. (Denna server kommer att göras tillgänglig under hösten 2010.)

(Även andra maskiner finns tillgängliga för enstaka kurser eller speciella ändamål.)

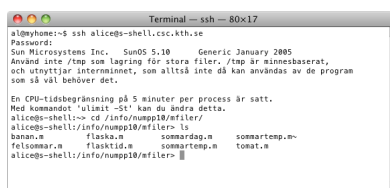
För att programmet ska kunna kryptera uppkopplingen, så behöver det serverns nyckel. De flesta program sparar nyckeln, så du behöver inte bekräfta den nästa gång du loggar in (med det programmet, på den datorn). Nyckeln för CSCs allmänt tillgängliga servrar finns på [Fjärrinloggning på CSC](http://www.kth.se/dokumentation/it-support-csc/faq/fragor-och-svar-kring-fjarrinloggning-pa-csc) (<http://www.kth.se/dokumentation/it-support-csc/faq/fragor-och-svar-kring-fjarrinloggning-pa-csc>)



```
al@myhome
File Edit View Terminal Help
al@myhome:~$ ssh alicegs-shell.nada.kth.se
Password:
Sun Microsystems Inc. SunOS 5.10 Generic January 2005
Använd inte /tmp som lagring för stora filer. /tmp är minnesbaserat,
och utnyttjar internminnet, som alltså inte då kan användas av de program
som så väl behöver det.

En CPU-tidsbegränsning på 5 minuter per process är satt.
Med kommandot "ulimit -St" kan du ändra detta.
alices-shell:~$ cd /info/numppi0/efiler/
alices-shell:~/info/numppi0/efiler$ ls
banan.n flaska.n somardag.n somartemp.n
felsomar.n flasktid.n somartemp.n tonat.n
alices-shell:~/info/numppi0/efiler$
```

Ubuntu (Terminal)



```
Terminal - ssh - 80x17
al@myhome:~$ ssh alicegs-shell.csc.kth.se
Password:
Sun Microsystems Inc. SunOS 5.10 Generic January 2005
Använd inte /tmp som lagring för stora filer. /tmp är minnesbaserat,
och utnyttjar internminnet, som alltså inte då kan användas av de program
som så väl behöver det.

En CPU-tidsbegränsning på 5 minuter per process är satt.
Med kommandot "ulimit -St" kan du ändra detta.
alices-shell:~$ cd /info/numppi0/efiler/
alices-shell:~/info/numppi0/efiler$ ls
banan.n flaska.n somardag.n somartemp.n
felsomar.n flasktid.n somartemp.n tonat.n
alices-shell:~/info/numppi0/efiler$
```

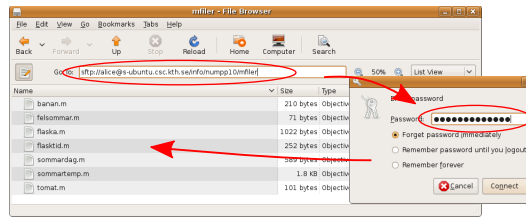
Mac OS X (Terminal.app)

7.5.2 Filöverföring (SFTP, SCP)

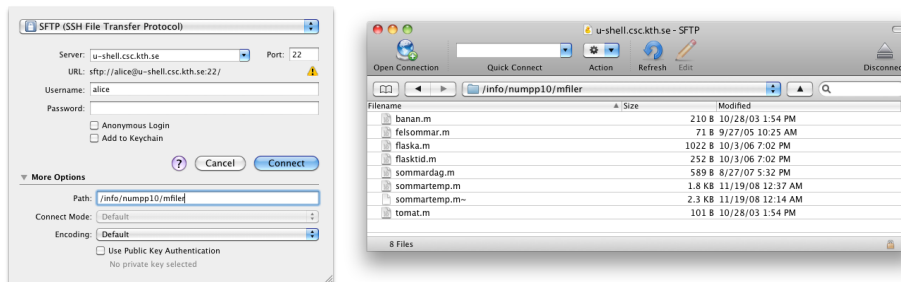
De flesta filhanterare i Linux känner till SFTP, så om man försöker öppna "katalogen" `sftp://«användare»@dator.csc.kth.se/«sökväg»` så dyker det upp en dialogruta där man kan skriva in användarnamn och lösenord. Sedan är det bara att öppna ett till fönster, och dra-och-släppa filer mellan fönstren.

¹⁵De traditionella programmen *Telnet* och *FTP* kan också användas, men det krävs att man har en version som stöder säkerhetssystemet Kerberos. SSH etc. rekommenderas istället.

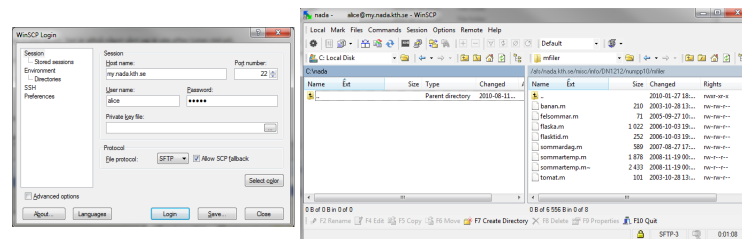
¹⁶*PuTTY* (<http://www.chiark.greenend.org.uk/~sgtatham/putty/>); programmet är vanligt nog att vara första träffen på Google.



I Mac OS X medföljer tyvärr ingen grafisk SFTP-klient. Man kan använda kommandoradsversionen, men det är förmodligen lättare att ladda hem och installera programmet *Cyberduck*¹⁷.



Till Windows, finns programmet WinSCP.



7.5.3 X11 forwarding

En av de största fördelarna med SSH är att man automatiskt får tillgång till *X11 forwarding*, dvs att program som körs på den dator man loggat in på, visas på den lokala datorn. Det fungerar ifall man har en X-server på den lokala datorn, något som finns på UNIX och Mac OS X, men måste installeras separat på Windows¹⁸

Har man en X-server, så behöver man normalt bara starta SSH med väljaren `-X`. Observera att det är en versal (stor bokstav), `-x` gör något annat.

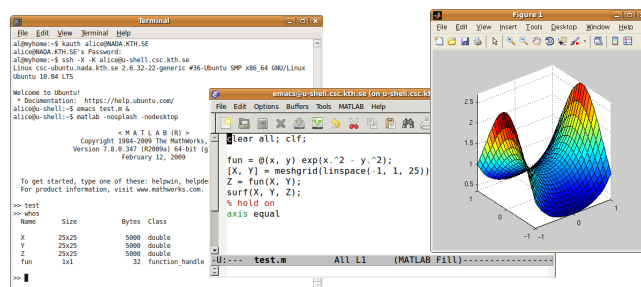
Har man inte en snabb uppkoppling, så kan många program upplevas långsamma, då all kommunikation sker över Internet. Ett program som Emacs, som inte gör mycket mer än att visa text, brukar fungera bra. För andra program kan man behöva stänga av så mycket grafiska finesser som möjligt. MATLAB går t ex att köra i rent kommandoradsläge, med väljarna `-nosplash -nodesktop`. För att skriva koden, kan man använda t.ex. Emacs¹⁹. Att visa plot-fönster fungerar, om än något långsamt.

¹⁷Cyberduck (<http://cyberduck.ch/>)

¹⁸Cygwin/X (<http://x.cygwin.com/>) är gratis, om än något krångligt att installera. Xming (<http://www.straightrunning.com/XmingNotes/>) är enklare, men endast äldre versioner är gratis.

¹⁹Är man van vid att använda Emacs, kan man för övrigt köra MATLAB som ett skal under Emacs. `M-x matlab-shell` startar MATLAB, och `C-C C-S` sparar filen, och evaluerar den.

7.5 Fjärrinloggning (ssh)



7.5.4 Tyngre och långvariga beräkningar

De CSC-datorer som studenter främst använder är salsdatorerna, och fjärrinloggningsservrar. Inget av dessa lämpar sig för att köra tyngre, långvariga beräkningar. Först och främst eftersom program som använder stora resurser (CPU-tid, minne, bandbredd) skapar problem för andra användares program. Men också därför att man (med salsdatorer) inte har kontroll över datorn. Om någon startar om datorn, så kommer beräkningarna att ha varit förgäves.

För långvariga beräkningar, så har CSC dedikerade beräkningsmaskiner, mer eller mindre kraftfulla datorer som finns i serverhallen. Dessa är dock inte allmänt tillgängliga. För att som student få tillgång till dessa, så måste man prata med sin kursledare.

Den som ska använda beräkningsmaskiner bör också sätta sig in i hur man bäst kör program på dem. Exempelvis behöver man vidta åtgärder för att program ska behålla Kerberos-biljetter när man loggar ut från maskinen. Man kommer förmodligen vilja använda program som `screen` och `kpagsh`, och lära sig hur man kör MATLAB utan grafiskt gränssnitt.

7.5.5 Fjärrinloggning med Kerberos

Ifall man vill slippa skriva sitt lösenord varje gång man loggar in, så kan man även installera Kerberos hemma. Det ger flera fördelar

- KDC²⁰ kan bekräfta din och serverns identiteter.
- *Single sign on* – du skriver ditt lösenord på *en* dator, för att skaffa biljetter. Andra datorer behöver aldrig se ditt lösenord, och kan därmed inte avslöja det.
- De nycklar som vidarebefodras har en begränsad livslängd. Stulna nycklar kan bytas, och blir då värdelösa.

De flesta moderna SSH-implementationer (inklusive SCP och SFTP) har med GSSAPI²¹ stöd för Kerberos. Likaså e-postklienter och webbläsare.

Vad gäller just SSH, så kan det användas för

Verifikation

Istället för att bekräfta en nyckel, så kan man låta Kerberos (med GSSAPI) bekräfta serverns krypteringsnyckel.

Autentisering

Ifall du har Kerberos-biljetter hemma, så kan de användas för att bekräfta din identitet; du behöver alltså inte skriva lösenord för att logga in.

²⁰Key Distribution Center, Kerberos-servern

²¹Generic Security Services Application Program Interface

Vidarebefodra nycklar

De biljetter du skaffar hemma kan göras vidarebefodringsbara (*forwardable*). Då skickas de till servern, och kan där användas för att skaffa ex AFS-tokens.

7.5.2**3.2.1**

Tillsammans med [inbyggt stöd i filhanterare](#) [○](#) för SFTP, och möjligheten att lägga upp [genvägar som automatiskt kör SSH](#) [○](#), betyder det att man efter att ha kört `kauth` kan öppna godtyckligt många nya terminaler och filhanterarfönster mot en CSC-dator, allt utan att skriva sitt lösenord igen.

(Det är dock bra att göra `kdestroy` innan man stänger datorn. Startar man om datorn i felsäkert läge kan det gå att komma åt biljetterna, och privata datorer är till skillnad från salsdatorerna oftast inte inställda på att förhindra felsäker omstart.)

För mer information om hur man installerar och ställer in Kerberos finns på Systemgruppens webbsidor: [Konfigurera Kerberos på UNIX/Linux](#) (<http://www.kth.se/dokumentation/it-support-csc/environment/solaris/instructions/kerberosconfig>).

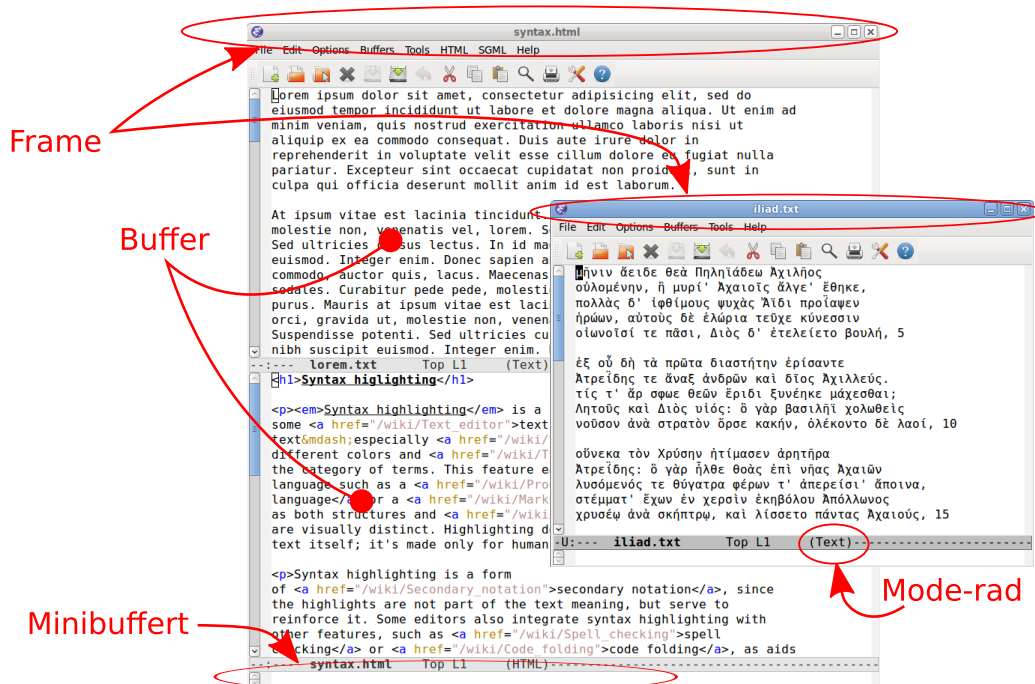
8 Emacs

Ska man skriva en enstaka textfil, så finns det lättare texteditorer att använda än Emacs. Ska man däremot skriva mycket text, vilket man oftast gör när man programmerar, så lönar det sig att lära sig använda någon editor som är utvecklad för programmering, och Emacs är en av de äldsta och vanligaste i UNIX-miljöer. Det är också det program som rekommenderats på CSC sedan många år tillbaka.

Emacs har funnits sedan 70-talet, och är så vanligt att även andra program (exempelvis kommandoraden) ofta använder dess vanligaste kortkommandon. Emacs styrka är dess stora bibliotek av funktioner – många ”hjälpfunktioner” är praktiskt taget egna applikationer.

Nackdelen är att alla finesser gör det svårare att lära sig använda programmet. Lättast är det, ifall man börjar med att använda menyerna, dvs **File** → **Open File...** för att öppna en fil, etc. Men programmet är uppbyggt för att användas med kortkommandon – endast en del av alla funktioner går att nå via menyer. Så, allt eftersom du börjar använda fler funktioner, försök också att lära dig fler kortkommandon. En kortare lista finns i slutet av detta avsnitt; en mer utförlig finns i slutet av detta häfte.

Emacs använder även en del ålderdomliga termer, som inte längre används på samma sätt i andra sammanhang. Ett fönster kallas för en *frame*, en fil kallas för en *buffer*, och eftersom det finns så många kortkommandon att man oftast måste använda två tangentkombinationer istället för en, så skrivs de C-x C-o istället för Ctrl-X Ctrl-O som annars är vanligare.¹

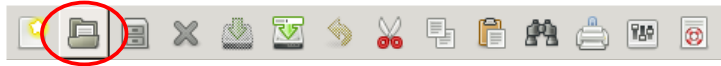


I Emacs kan du i allmänhet utföra samma kommando på flera olika sätt. Till exempel, att öppna en fil kan göras genom att:

1. Välja **File** → **Open File...**, som visar en grafisk dialog där man kan välja filen. **Visit New File** kan användas ifall man vill skapa en ny fil: Om man ”öppnar” en fil som inte finns, så skapas den när man sedan sparar bufferten.

¹I detta häfte används normalt det senare, men i detta avsnitt går vi över till att använda det som finns i Emacs dokumentation.

2. Välja motsvarande motsvarande knapp i verktygsraden, under menyerna.



3. Använda kortkommandot `C-x C-f` och skriva in namnet i Emacs minibuffert (raden längst ner). Man kan, precis som i terminalfönstret använda «tab» för att komplettera.
4. Använda `M-x` och med namn anropa funktionen *find-file*. Tryck `M-x`, skriv *find-file* («tab» fungerar), och tryck «retur».

För de funktioner som finns i menyerna, så står kortkommandot brevid menyvalet, och man får en kortfattad beskrivning ifall man håller musen stilla över det.

Ifall man skriver fel när man ska skriva den ibland ganska krångliga sekvens² som krävs för ett kortkommando, så kan man avbryta genom att en eller ett par gånger trycka `C-g`.

8.1 Textredigering, klipp-och-klistra

Markören, en blinkande svart rektangel, visar var *insättningspunkten* är, den plats där texten du skriver hamnar. För att flytta den kan man givetvis använda vanliga piltangenterna och «Home», «End», etc. För att man ska slippa flytta händerna på tangentbordet, finns dock även kortkommandon för det

`C-a` Flyttar till början av raden

`C-e` Flyttar till slutet av raden

Radera text gör du bakåt med «backsteg», och framåt med «delete» eller `C-d`. Man kan även använda:

`C-k` Radera resten av raden (*k* för *kill*)

`M-d` Radera nästa ord


`C-w` Radera markering (klipp ut)

`M-w` Kopiera

`C-y` Klistra in (`S-«insert»` fungerar också)

`C-_` Ångra redigering.

Markering av text görs genom att man trycker `C-«space»` för att starta en markering, flyttar markören, och sedan klipper/kopierar med `C-w` eller `M-w`. Att klistra in görs sedan med `C-y`³

[2.1](#) Man kan även markera med musen. Emacs använder sig av samma kopiering som att [mitten-klicka](#)  med musen. Alltså, ifall man markerat text med första musknappen, kan man klistra in det i Emacs med `C-y`.

²Minibufferten visar, efter ett par sekunder, vad man tryckt för kombination.

³Alla urklipp läggs i en s.k. *kill ring*. Man kan bläddra till tidigare urklipp med `M-y`.

8.2 Söka text

För att söka efter ett visst ord, placerar du markören där du vill starta sökningen, och trycker `C-s`. Skriv det ord du söker efter, så går Emacs dit (och markerar andra platser där ordet finns). För att hitta nästa förekomst, tryck `C-s` igen. För att hoppa ur sökningen, tryck `«retur»`⁴. För att återgå dit du började söka, tryck `C-g`.

Sökningen ”minns” vad man sökte efter sist, så har man sökt ett ord tidigare, kan man trycka `C-s C-s` för att söka igen efter samma sak. `C-r` söker bakåt, i övrigt på samma sätt.

”Sök-och-ersätt” finns som **Edit** → **Replace** → **Replace String...**, eller `M-%`. Emacs frågar för varje förekomst ifall den ska ersättas. `y` ersätter, `n` hoppar till nästa, och `!` ersätter samtliga (kvarvarande) förekomster.

8.3 Hantera flera filer

Emacs kan ha flera aktiva filer, varje i sin egen buffert. Du kan välja vilken du ser från menyn **Buffers**, även om det vanligaste är att använda `C-x b` för att byta buffert.

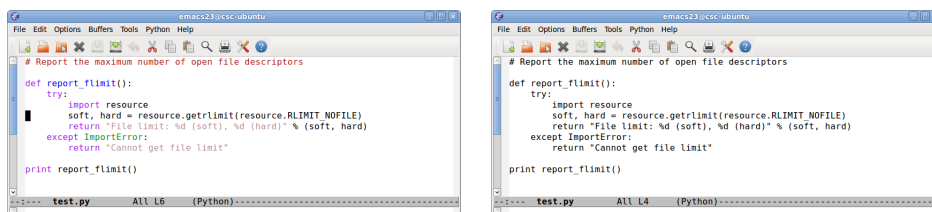
Du kan också dela upp Emacs i flera olika underfönster. **File** → **Split Window** eller `C-x 2` delar på höjden, `C-x 3` på bredden och `C-x 1` gör aktuell buffert till den enda i fönstret. Ett nytt fönster får man med **File** → **New Frame**, `C-x 5 2`.

Ifall aktuell fil inte sparats på ett par minuter, så sparar Emacs en kopia med `#` tillagt, så att `hello.c` blir `#hello.c#`. Råkar man logga ut utan att spara, så finns den kvar, och kan användas för att återskapa filen.

Om du har öppnat en befintlig fil för redigering, så sparas dessutom en ”tilde-fil”, dvs en fil med ett `~`-tecken efter (`hello.c~`). Oavsett hur många gånger du sparar filen, så är `~`-filen en kopia av det som fanns innan du öppnade filen.

8.4 Lägen

Emacs uppträdande ändras beroende på vilken fil som redigeras. Detta styrs av olika *lägen*. Detta styrs normalt av de filändelser som Emacs känner igen, men det går också att ändra manuellt. Det varierar mycket vad som ändras när Emacs byter läge, men normalt är att några menyer läggs till, vissa tangenter får ändrad betydelse, och att syntaxfärgning (eng *syntax highlighting*) används för att färglägga vissa ord i texten efter deras betydelse. Aktuellt läge visas i *modraden*.



Syntaxfärgning (*font-lock-mode*) på resp av.

För att manuellt ändra läge, använder man exempelvis `M-x matlab-mode`.

⁴Vill man söka efter radbrytning, trycker man `C-j`

8.5 Inställningar

Emacs sparar inställningar i filen `~/ .emacs`. Tidigare var man tvungen att manuellt redigera denna fil, och många hjälpsidor på nätet ger fortfarande instruktioner för att göra det. (Det nyare sättet är trots allt bara 15 år gammalt. . .) Man kan bläddra igenom inställningarna i grupper genom att välja **Options** → **Customize Emacs** → **Top-level Customize Groups**.

Alternativt kan man ändra en enda inställning, genom att i samma meny välja **Specific Option...** (eller `M-x customize-option`), och skriva in den variabel inställningen styr. Variablerna heter samma sak som inställningarna, men med alla bokstäver ersatta av gemener och bindestreck istället för mellanslag. Alltså, *Line Number Mode* styrs av `line-number-mode`.

Några av de vanligaste valen är

Global Font Lock Mode Syntaxfärgning (*syntax highlighting*), dvs använda färger för att göra kod mer lättläst. Mycket praktiskt när man vant sig vid det. Idag är det påslaget per default, tidigare var man tvungen att slå på det under **Options** → **Syntax Highlighting**.

Transient Mark Mode Visa aktuell region med avvikande stil (beige bakgrund). Gör det lättare att klippa-och-klistra med kortkommandon.

Show Paren Mode Markerar matchande parentespar (eller hakparenteser, klamrar). Även på **Options** → **Paren Match Highlighting**.

CUA Mode Genom `M-x cua-mode`, eller motsvarande val under **Options**, ställer man in Emacs kortkommandon till att följa *Common User Access* riktlinjer. I praktiken, så använder Emacs då `Ctrl-X`, `Ctrl-C` och `Ctrl-V` för att klippa ut, kopiera och klistra in. I de flesta fall fungerar övriga kortkommandon som `C-x C-s` normalt (man kan inte klippa om man inte har en aktiv region), och det går även att använda `Shift` samtidigt, dvs `S-C-x C-s`, för att komma åt de vanliga kortkommandona.

8.6 Hjälp

Emacs har inbyggd dokumentation, som är mycket utförlig. Menyn **Help** innehåller bland annat *Emacs tutorial*, som ger en mer grundläggande introduktion till programmet. Hjälp om kortkommandon, lägen osv finns under **Help** → **Describe**. Väljer man där *Describe Key* innan man utför ett kortkommando eller klickar på ett menyval, så utförs inget. Istället visas dokumentationen för det som normalt skulle ha utförts. *List Key Bindings* ger en (lång!) lista på vilka kortkommandon som finns.

8.7 Alternativ

Det finns ett par olika alternativ till Emacs, som det kan vara bra att känna till.

XEmacs Efter snart 35 års utveckling finns det många olika versioner av Emacs, förutom den vanliga (formellt kallad GNU Emacs). En av de vanligaste varianterna heter *XEmacs*. Kortfattat, så märks skillnaderna först när man är van vid en version, och funktionalitet som införs i den ena, brukar tämligen snart dyka upp i den andra.

Utah X? Ifall man av någon anledning inte har tillgång till en fönstermiljö (oftast när man startar en dator i *single user mode* för att felsöka) så finns sällan Emacs tillgänglig. "Ersättare" är vanligen *nano* eller *pico* under Linux, eller *mg* under Solaris.

9 Programmering

Programmering tas upp i de olika kurserna på CSC. Där får man lära sig vad ett program är, och hur man skriver den kod som ett program byggs upp av. Eftersom detta görs på olika sätt beroende på vilket programspråk man använder¹ (och olika kurser använder olika språk), så tar detta häfte inte upp de detaljerna. Vi tar istället upp hur man rent praktiskt ska köra sitt program.

Även om du på dina kurser använder olika program som gör en del av arbetet åt dig, så är det bra ifall du även sätter dig in i hur man kan klara sig utan dem. Har man bara lärt sig att klicka på knappen "Kör" i en IDE (*Integrated Development Environment*), så är det svårt att se skillnad på kompilering och exekvering. Har man däremot lärt sig att använda *java* och *javac*, så förstår man vad som händer när man klickar på knappen.

Sammanfattningarna nedan är inte tänkt som en förklaring; man måste ha lärt sig begreppen tidigare för att följa med.

9.1 Java

Ett Java-program skrivs i en (eller flera) filer med ändelsen `.java`. Det är ett *objektorienterat* språk; varje fil kan innehålla en eller flera *klasser*. Normalt använder man en fil till varje klass för större projekt, men ifall klasserna är små och inte så många, så kan man ibland lägga dem i samma fil².

Varje fil *kompileas* med programmet *javac* till binär kod, en (eller flera) *klassfiler*. Man går från `«något».java` (text) till `«något».class` (binär).

Eftersom Java-program är plattformsoberoende³, så använder man programmet *java* för att köra sitt program. Det finns även *applet*:ar, program som visas i en webbläsare eller med programmet *appletviewer*⁴. Tekniskt säger man att programmet *tolkas* av en *virtuell maskin*. Begrepp som *just-in-time*-kompilering gör dock saken mer komplicerad.

Exempel

```
> javac HelloWorld.java
> ls HelloWorld.*
HelloWorld.class      HelloWorld.java
> java HelloWorld
```

För en applet, behöver man förutom sin *class*-fil även en HTML-fil med ett `<applet>`-element i.⁵ Den kan visas i en webbläsare, eller genom

```
> appletviewer HelloWorld.html
```

¹Om man lär sig flera programspråk börjar man ganska snart se gemensamma tankar och uttrycksätt – en if-sats fungerar ungefär likadant oavsett om det är i C, Python eller MATLAB. Men i början är det svårt att se skogen för alla träd, dvs skilja på syntax och begrepp.

²Det finns regler för när detta går eller inte går att göra.

³Samma binära kod kan köras på olika datorarkitekturer eller operativsystem.

⁴Även om det går att skriva ett Java-program som kan köras antingen som program eller som applet, så är detta svårt. Det är bättre att du ordentligt lär dig respektive sätt att utveckla program, och först sedan ser på hur man kan slå ihop det.

⁵Att använda den kortare formen av ett element, `<applet [...] />` istället för `<applet [...]></applet>`, rekommenderas ej då inte alla applet-visare stödjer det. Det är också bättre att använda *applet*-elementet, även om det anses föråldrat (*deprecated*) så stöds det bättre än det nyare *object*.

9.1.1 Integrerad utvecklingsmiljö, IDE (*Eclipse*)

Det finns ett par olika integrerade utvecklingsmiljöer för Java-program. En IDE innehåller förutom texteditor verktyg för att organisera de olika filer som bygger upp programmet, för att kompilera och köra koden, en debugger, osv. Eclipse innehåller dessutom verktyg för att göra grafiska gränssnitt.

Du hittar Eclipse under **Applications** → **Programming**.

9.2 Python

Python är ett programspråk som bland annat är byggt för att man ska kunna skriva läsbar kod. Till skillnad från andra programspråk är indenteringen (vänstermarginalen) en del av språket, inte bara något som man justerar för att göra koden mer läsbar.

Python är byggt för att stödja många olika paradig⁶, inklusive objektsorientering och (i viss mån) funktionell programmering. Version 3 av språket är *inte* bakåtkompatibel med version 2, även om vissa av nyheterna införts även i den versionen (s k "backporting").

Bägge versioner finns installerade, som `python2` (vilket är default) resp `python3`:

```
> python2
Python 2.6.5 [...]
>>> print round(1.5), " :: ", round(2.5)
2.0  ::  3.0
>>> exit()


> python3
Python 3.1.2 [...]
>>> print(round(1.5), " :: ", round(2.5))
2  ::  2
>>> exit()
```

9.2.1 IDE (*Idle*)

Python distribueras tillsammans med en tämligen rudimentär utvecklingsmiljö, IDLE. Den körs normalt med ett fönster för textredigering, och ett för in- och utmatning till programmet.

Du hittar Idle under **Applications** → **Programming**.

[5.1.2](#)


Ifall du kör Idle i en tunn klient under [Solaris](#) , kan du behöva använda väljaren `-n`.

9.3 MATLAB

MATLAB (*Matrix Laboratory*) används för matematiska beräkningar som bygger på numeriska (snarare än symboliska) metoder. MATLAB räknar med matriser av tal, och matrismanipulationer utgör en grundläggande del av språket. Programmet används ofta för att visualisera data på olika sätt, med grafer i 2D och 3D.

Programmet har sin egen inbyggda IDE, men det går även att köra utan den⁷ Miljön fungerar i stort sett likadant på olika plattformar, men ifall du

⁶Olika (programmerings)paradigm är olika sätt att organisera och strukturera program. Vilket paradigm som är lämpligt beror ofta på problemet.

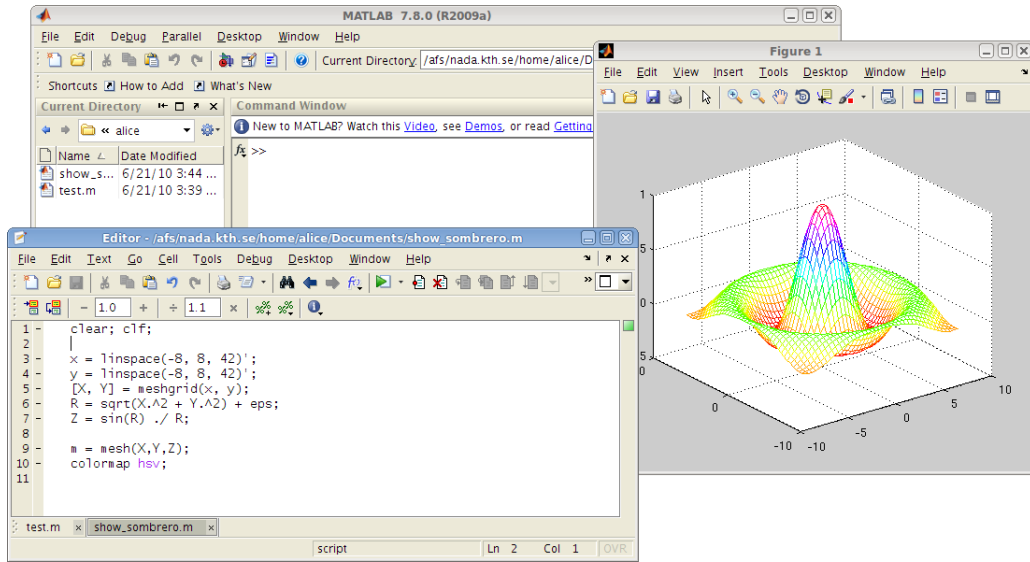
⁷Behövs förmodligen inte på grundkurser, men kan vara praktiskt ifall man ska göra [tyngre beräkningar](#)  [\[7.5.4\]](#).

9.3 MATLAB

är van vid exempelvis Windows, kan du vilja ändra så att dess kortkommandon används även under UNIX, där man annars använder "Emacsliknande" kortkommandon. Inställningen görs under **File** → **Preferences...** → **Keyboard**, kallas *Command Window key bindings* resp *Editor/Debugger key bindings*.

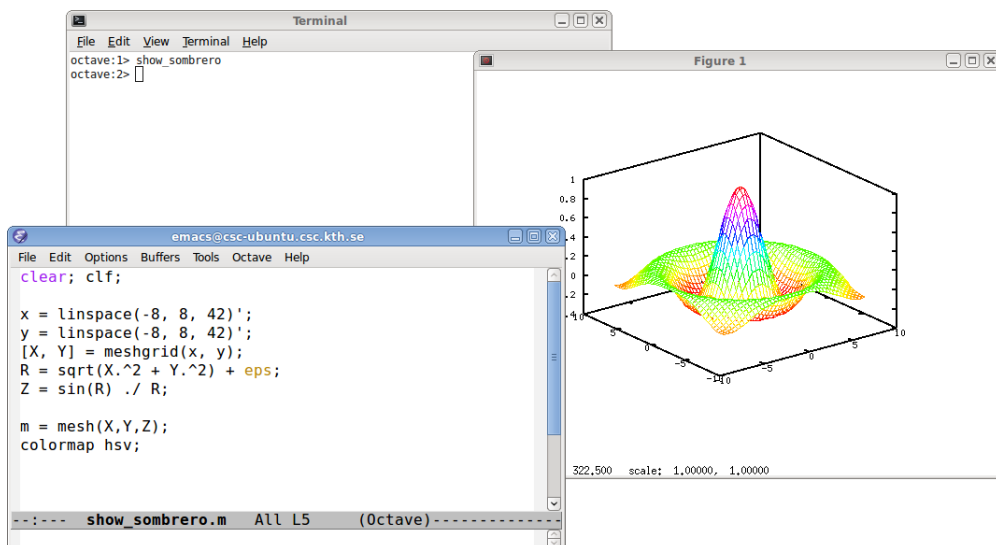
Notera också att MATLABs kommandofönster går att använda som en riktig terminal. `ls` och ett par andra kommandon är inbyggda funktioner i MATLAB, men du kan även anropa andra kommandon som `!«kommando»`.

Du hittar MATLAB under **Applications** → **Programming**.



9.3.1 Octave

GNU Octave är, likt MATLAB, gjort för att användas för numeriska beräkningar. Dess språk är i hög grad inspirerat av MATLABs, och med lite vana är det inga större problem att skriva program som kan köras i endera Octave eller MATLAB. För den som inte är van att programmera, kan dock skillnaderna vara förvirrande.



Du hittar (GNU) Octave under **Applications** → **Programming**, men det går lika bra att köra det med kommandot `octave` i ett terminalfönster (genvägen

under **Applications** kommer i annat fall att starta ett nytt). För att skriva koden, använd valfri texteditor.

9.4 Subversion (svn)

Apache Subversion (svn) är ett verktyg för versionshantering som kan användas för mjukvaruutveckling, dokumentation, eller annan information. Subversion kan hantera både text- och binärfiler, och låter dig behålla och arbeta med flera dokument eller olika versioner av dokument samtidigt. För mer information om hur man använder svn på CSC, se <http://svn.csc.kth.se>.

10 Dokumentation och hjälp

Ingen dokumentation kan täcka in allt. För att hitta information om datorsystem och program, så behöver man alltså lära sig hur man hittar information, snarare än att använda ett specifikt hjälp-program. Det mesta går att hitta på webben, men för att förstå dokumentationen kan man behöva en del grunder. Dels för att veta några termer man kan specificera sitt sökande med, men också för att lära sig *hur man läser dokumentation*. Dokumentation är oftast gjord som uppslagsverk snarare än lärobok. Jämför med matematik: om en ingenjör behöver derivera \arcsin så vill han inte lära sig derivera, eller härleda de trigonometriska funktionerna. Det känner han redan till. Han vill bara veta att

$$\frac{d}{dx} \arcsin x = \frac{1}{\sqrt{1-x^2}}$$

Man-sidor (eller annan hjälp) är på samma sätt gjord främst för de som i huvudsak vet vad de vill göra, men behöver slå upp någon detalj. Vet man att `ls` listar filer, så kan man slå upp att väljaren `-t` gör så att filerna sorteras efter de datum de senast modifierades.

När man på så vis är ute efter en specifik detalj, så behöver man veta hur dokumentationen är organiserad. Då går det mycket snabbare att skumma igenom för att hitta det man behöver.

10.1 Man-sidor (man, info)

Man-sidor (för "manualsidor") är det snabbaste sättet att hitta information om program och inställningar på systemet. För att visa en man-sida i terminalen används

```
> man sökord
```

Sidan innehåller¹ först en kort sammanfattning av vad programmet gör, sedan en sammanfattning (*synopsis*) med olika sätt att anropa programmet. Sedan kommer en mer förklarande beskrivning, som också berättar vad de olika argumenten är.

I det här fallet ska det läsas som att man i första fallet ska skriva `mv`, man kan, om man vill, använda väljaren `-f` eller `-i`, man ska ha något som kallas "source", och något som kallas "dest", men den faktiska texten ska bytas ut mot något annat under anropet.

I det andra fallet (alltså ett annat sätt att anropa `mv`), så kan man dessutom använda väljaren `-u`. Eftersom det kommer "..." efter "source", så får det nu upprepas (man kan ange flera filer). Nu ska sista argumentet vara "directory".

Under rubriken *description* beskrivs vad argumenten ("source", etc) är, men med lite vana förstår man redan av sammanfattningen att man det första fallet är vad man gör när man byter namn på en fil, och det andra är när man flyttar in flera filer i en katalog. Väljarna förklaras också, i det här fallet `-i` för att fråga innan man skriver över, och `-f` för att aldrig fråga.

MV(1)	User Commands	MV(1)
NAME	<code>mv - move (rename) files</code>	
SYNOPSIS	<code>mv [-fi] source dest</code> <code>mv [-fiu] source... directory</code>	
DESCRIPTION	[...]	

Figur 2: man-sida för `mv`

¹Den faktiska sidan ser något annorlunda ut, med fler väljare

Skrivsättet kan variera litet. Det skulle kunna stå "SOURCE", "source" eller "<source>" för att markera det som ska bytas ut. Men i huvudsak är man-sidor ganska konsekventa.

Tips om närliggande kommandon brukar finnas under rubriken *see also* vid slutet av sidan. Man-sidor delas in i katagorier. Ifall man säger "se *mkdir(1)*", så menar man den *mkdir* som är ett kommando, inte *mkdir(2)*, en systemfunktion i programspråket C.

Kommandot `apropos` kan också användas för att få tips om program:

```
> apropos rename
mv (1) - move (rename) files
pts_rename (1) - Changes the name of a Protection Database
entry
rename (1) - renames multiple files
vos_rename (1) - Renames a volume
```

Man-sidor finns på nätet, både på CSCs webbsidor och på andra platser. Men detaljer såsom väljare kan variera något mellan olika operativsystems versioner², så ifall man kan är det säkrast att titta på den dator man använder.

Info-systemet är ett något annorlunda dokumentationssystem. Det är mer uppbyggt kring hyperlänkad³ ("klikbar") dokumentation än vad man-sidor är. Infodokument används främst av dokumentation till GNU-program. De kan läsas på webben, i texteditorn Emacs, eller med kommandot `info`.

10.2 Online-hjälp

Specifikt för Ubuntu, så har Canonical, Ubuntu's skapare, länkar till gratis dokumentation, skapad av användare, på [sin hemsida](http://help.ubuntu.com) [〈http://help.ubuntu.com〉](http://help.ubuntu.com). Det finns också ett [forum](http://ubuntuforums.org/) [〈http://ubuntuforums.org/〉](http://ubuntuforums.org/), och dokumentation [på svenska](http://ubuntu-se.org) [〈http://ubuntu-se.org〉](http://ubuntu-se.org).

För andra system finns många FAQ och webbsidor som kan svara på frågor. Länkarna i detta häfte till Wikipedias artiklar på ett flertal ämnen, kan ge en bra plats att starta på. De ger oftast information om vilka som utvecklat systemet, och länkar till deras hemsida.

Mycket av informationen är på engelska, men Wikipedia ger också länkar till svenska artiklar, i den mån de finns. Det är dock långt ifrån all programvara som har svensk dokumentation.

10.3 Mänsklig hjälp: handledning

10.3.1 Kurshandledning

För alla kursrelaterade frågor – övnings- och inlämningsuppgifter, labbar, etc – så är det alltid bäst att fråga övningsledare, handledare eller kursledare. De är bekanta med vilka krav som gäller, vilken litteratur som finns, teorin bakom, o s v. Vissa kurser har schemalagda tider för frågor, andra svarar på dem under föreläsningar eller övningar. Se kurshemsidan för detaljer.

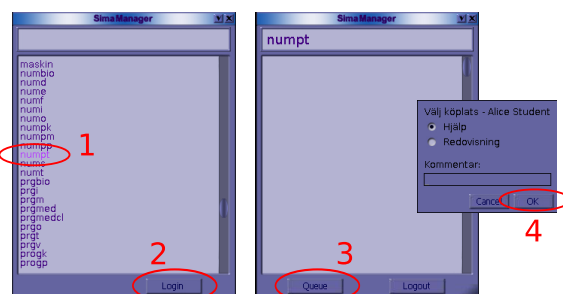
²Exempelvis mellan CSCs olika Solaris- och Linux-plattformar.

³"Hypertext" är text som med "hyperlänkar" är förbunden med andra texter. Man behöver därför inte läsa linjärt, utan kan hoppa fram och tillbaka. Webbsidor, som bygger på protokollet HTTP, *HyperText Transfer Protocol*, är ett exempel.

10.4 Problemlösning – FAQ

10.3.2 Sima Manager

Många kurser använder den datoriserade kölistan *Sima Manager* på datorlaborationerna. Du hittar Sima under **Applications** → **CSC** → **Sima Manager**.



Starta programmet, välj rätt kurs och **Queue** för att köa. Ifall du använder en bärbar dator, skriv in var du sitter i kommentarsfältet!

10.3.3 Allmänhandledning

Allmänhandledaren är tillgänglig två gånger om dagen för att ge hjälp med allehanda programmerings- eller datorrelaterade problem. Observera att kursspecifika frågor i regel *inte* kan besvaras, då allmänhandledaren inte kan vara insatt i kraven för alla kurser. Å andra sidan, så har allmänhandledarna erfarenhet av många kurser, och kan ofta ge en välgrundad gissning angående vad som söks.

Det är endast en handledare på varje pass; om ett problem tar lång tid att svara på kan du behöva vänta länge på din tur. Assistenten är alltid på 4:e våningen. För att få hjälp, använd Sima (ovan).

[Allmänhandledningens hemsida](http://www.nada.kth.se/handledning/) (<http://www.nada.kth.se/handledning/>) har ett schema, som alltid hålls uppdaterat, och även ett FAQ med vanliga frågor.

10.3.4 Systemgruppen

CSCs datorer administreras av Systemgruppen. Problem med datorerna bör rapporteras till dem, på system@csc.kth.se.

Även om de inte kan svara på kursrelaterade frågor, så kan de kanske svara på andra datorfrågor. Delfi, deras help desk, är öppen alla veckodagar, och epost till adressen ovan läses kontinuerligt under kontorstider. Se [deras hemsida](http://www.kth.se/csc/it-support-csc) (<http://www.kth.se/csc/it-support-csc>) för öppettider.

10.3.5 Annan datorsupport


KTH har en sida med [länkar till de olika datorsupportgrupperna](http://www.kth.se/student/support?l=en_UK) (http://www.kth.se/student/support?l=en_UK).

10.4 Problemlösning – FAQ⁴

Fil raderad av misstag

Se [säkerhetskopior](#) , den kanske finns i OldFiles? [7.3.2](#)

Filer är tomma

Ifall du har fullt på kontot så kanske ett program inte kunde spara en öppen fil? Se [kvot](#) . [7.2.2](#)



⁴Frequently Asked Questions, vanliga frågor

Firefox ger "konstiga" problem

Kan också bero på fullt konto. *Firefox* betar sig ofta konstigt ifall den inte lyckats spara alla inställningar. Ibland blir SSL-inställningarna korrupta, då kan man inte öppna någon säker webbsida (ex *Mina Sidor* under KTH.SE).

Felmeddelande om "over quota"

7.2.2

Lagringskvoten är överskriden. För att se vad som tar upp plats, kan du använda *Disk Usage Analyzer* (Baobab). Se [kvot](#)  och [exemplet med Baobab](#) .

3.2.1

- Prova att rensa bort onödiga filer.
- Arkivera och komprimera filer du inte behöver så ofta. Ubuntus *Archive Manager* heter *file-roller* och körs i terminal eller med Alt-F2.
- Ifall det är kursmaterial som gör att du behöver extra minnesutrymme, kontakta CSCs datorsupport, eller din kursledare.


Jag bytte lösen, men det nya fungerar inte

Har du bytt rätt lösenord? Det finns många olika konton på KTH, och ofta har de samma användarnamn. För att använda CSCs datorsystem, så behöver du ett konto på CSC. Detta konto är inte samma konto som det centrala KTH.SE-konto som används på *Mina Sidor*.

E-post till CSC-kontot («användare»@csc.kth.se) är oftast inställd på att vidarebefodras till «användare»@kth.se. Detta är en inställning du kan ändra själv (med kommandot `chpobox`), men det är bra att vidarebefodra ena kontot till det andra, så att du uppmärksammar eventuella automatiska meddelanden.


Kan inte logga in

3.1.1

Fel lösenord? Det finns som sagt många olika konton, så det är lätt att blanda ihop dem. Eller så kanske det är något så enkelt som *Caps Lock*? Prova på en annan dator, eller med [felsäker inloggning](#) .

10.5.4

Om det fungerar på en annan dator, så felanmäl gärna den dator du inte kunde logga in på.

Ifall det fungerar med felsäker inloggning, så kan du [återställa default-inställningarna](#) .

Trög dator


6.3.11

Förmodligen tar en eller flera processer upp datorns processorkraft eller arbetsminne. Se [processhantering](#) .

Ifall en annan användares processer ger dig problem, kontakta datorsupporten. Oavsett om det beror på okunskap eller ohyfs, så bör det åtgärdas!

7.2.1

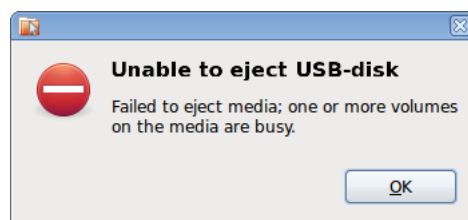
Kan inte spara/läsa vissa filer

Ifall du varit inloggad över 10 timmar, kan dina [Kerberos-biljetter](#)  ha slutat gälla. Kör `kauth` i terminal.

3.7

Kan inte ta bort USB-minne

Ifall du försöker ta bort ett [USB-minne](#)  och får meddelandet



så använder något program den katalogen. Du kanske har en terminal öppen, där du står i den katalogen?

10.5 För dig som har ett gammalt konto

För den som fått ett nytt datorkonto på CSC så ska allting ”bara fungera”, åtminstone tills man börjar göra egna ändringar av sina inställningar. Men för den som av någon anledning⁵ har ett konto som skapats för flera år sedan, så finns det ett par saker man kan vilja ändra på.

För detaljer om *hur* man löser dessa problem, kontakta CSCs datorsupport.

10.5.1 Konstigt användarnamn

Fram till ungefär år 2000, så var dåvarande Nadas användarnamn av formen «*grupp*»«*år*»-*abc* eller «*grupp*»«*år*»_*abc*, exempelvis *f99_aan* för Alice Andersson. Ett par år därefter användes ”u1-namn”, på formen *u1ak8b73*, dvs den unika identifierare som används i KTHs centrala databas⁶. Idag används samma användarnamn som på det centrala KTH.SE-kontot.

För den som så önskar, kan gamla konton döpas om (till samma namn som KTH.SE-kontot).

10.5.2 ”Fel” skal (*tcsh* eller *bash*)

Under 2008 byttes default-skal från *tcsh* till *bash*.⁷ För den som inte själv satt sig in i skillnaderna spelar det förmodligen ingen roll vilket man använder. Det mesta i detta häfte fungerar likadant i både *tcsh* och *bash*.

Den som så önskar, kan dock byta med programmet *chsh* (**change shell**).

10.5.3 Gamla inställningar

Gamla inställningar kan skapa problem med det mesta. Ifall man tidigare haft konton på Elektro kanske man inte kan logga in eftersom man har en *.k5login* eller *.TheseCells* med felaktiga inställningar.⁸ Man kan få konstiga felmeddelanden, eftersom man försöker starta en Zephyr-tjänst som inte funnits sedan 2006.

Många kör den gamla fönsterhanteraren CDE i Solaris, trots att den nyare Java Desktop både ser nyare ut och upplevs modernare⁹.

Inställningar görs ofta (men inte alltid) i filer som heter «*något*»*rc* (för *run commands*), exempelvis *.bashrc* för *bash*.

De viktigaste inställningarna kan återställas, se nedan.

10.5.4 Återställa default-inställningar

För att återställa inställningar, så behöver du oftast logga in antingen med fjärrinloggning (SSH, se avsnittet om [fjärrinloggning](#) ☺), eller i en [felsäker session](#) ☺ (genom att välja **Failsafe** från login-skärmens sessions-meny). Annars riskerar dina nuvarande (felaktiga) inställningar att sparas igen,

[7.5](#)

[3.1.1](#)

⁵Man kan t.ex. ha gått kurser på CSC, eller tidigare Nada, för många år sedan. Ett sådant konto har normalt varit avaktiverat, dvs man har inte kunnat logga in, men så länge alla filer finns kvar, så gäller de inställningar man hade när kontot återaktiveras.

⁶Egentligen är det *u1xxx* som är ditt KTH-id, men så gott som samtliga KTH-system kan nu använda användarnamnet istället.

⁷Namnen kommer från historiska orsaker, blandat med ordvitsar. *tcsh* = TENEX C Shell, efterföljare till ett skal med syntax som liknar programspråket C, inspirerat av ett äldre operativsystem. *bash* = Bourne Again Shell, efterföljare till det klassiska *Bourne Shell*, *sh* som gjordes av Stephen Bourne

⁸Om du inte vet hur de fungerar, ska du nog inte ha någon av filerna. Har du en *.k5login*-fil, så *måste* den exempelvis peka ut din Kerberos-identitet på CSC.

⁹Andra kör fullt medvetet CDE, eftersom *if it ain't broke, don't fix it*.

när du loggar ut. Följdaktligen är de flesta verktyg för återställning gjorda för kommandoraden, och har inget grafiskt gränssnitt.

När du har loggat in till en kommandorad, använd `clear-env` (utan argument) för att se vilka inställningar du kan återställa, och hur. `clear-env` visar också vilka andra kommandon som kan användas för att återställa specifika inställningar.

11 Kursadministration

11.1 Mina Sidor

Administration av kurser, inklusive framtagande av studieintyg och tentaanmälan för kurser som tillämpar det, sker på *Mina Sidor*, (<http://www.kth.se/student/minasidor>). För mer information, se webbsidorna.

11.2 rapp

Rapp är ett system som används för *intern* kursadministration på CSC. Labb- och tentaresultat och liknande lagras i rapp, och du kan själv logga in för att se dina resultat. Dokumentation finns på webbsidan, (<http://rapp.nada.kth.se>).

11.3 course

Kommandot *course* används för att gå med i en kursomgång. Du får meddelanden från kursledaren, och får även automatiskt eventuella [moduler](#) 7.1.8 som kan behövas för kursen. (Förutsatt att din kursledare har lagt till dem i kursmodulen.)

Att använda *course*:

```
> course join numpp10
> course leave numpp10   ### efter kursen
> course list             ### lista mina kurser
```

Kommandot har också en del verktyg för att t ex dela filer (labbar o dyl) med andra användare:

```
> course labdir numpp10 alice bob charlie
```

11.4 res

Res är ett äldre system som under många års tid använts för intern kursadministration. I allt större utsträckning används nu *rapp*. *Res* är terminalbaserat, och för att använda det behöver du logga in på en Solaris-maskin. För att registrera sig för kursomgången *numpp10*, skulle du kunna göra:

```
csc-ubuntu:~> ssh s-shell
s-shell:~> res checkin numpp10
[...]
s-shell:~> res show numpp10
```

(*res show* fungerar endast efter att nyregistrerade rapporterats in av kursledare eller handledare.)

Bilagor

A Några termer


exekvera (utföra, eng *execute*) Att köra ett program. En applikation eller ett kommando består typiskt av ett eller flera program. Ett program är något förenklat att betrakta som en serie instruktioner, som utförs i tur och ordning.

klient, server När en dator eller ett program utför tjänster med hjälp av en annan dator (eller ett program), så kallas den som ber om tjänsten för "klient", och den som utför den "server" eller "värddator". Exempelvis är ett e-postprogram en klient till en server som hanterar e-posten.

En "server" kan också vara en (ofta kraftfull) dator som man inte använder lokalt, utan fjärransluter till.

plattform Beror på sammanhanget. Ofta en kombination av den hårdvara och mjukvara som används, exempelvis operativsystem och processtyp hos en dator. Exempelvis Mac OS X och Ubuntu Linux, men lika gärna Ubuntu på en Intel-x86-kompatibel processor och Ubuntu på en UltraSPARC-processor.

arbetsstation (eng *workstation*) Traditionellt sett en kraftfull dator som används för tekniska eller vetenskapliga applikationer, oftast ansluten till ett datornätverk. Idag är datorerna så kraftfulla att det mest syftar till hur man använder datorn. En "mediator" som spelar upp film och musik kan mycket väl vara mer kraftfull än den "arbetsstation" man använde för att skriva dess programvara.

GUI Grafiskt användargränssnitt (eng. *Graphical User Interface*). Används både om begreppet i stort¹ och om specifika GUI:n, "Programmets GUI är...". Se [GUI](#) .

tecken Namn på några vanliga tecken finns i bilaga B

whitespace Bokstäver som inte skrivs ut (dvs mellanslag, tabbar, radbrytning, etc) kallas för *whitespace*, efter engelskans *whitespace character*.


autentisera (även "autenticera"; eng *authenticate*) I säkerhetssammanhang: Bekräfta sin identitet. Alltså, visa att du verkligen är den användare du utger dig för att vara, vanligen med ett lösenord eller någon annan form av kod.

"foo?" I mindre formell dokumentation används ofta nonsens-ord som platshållare när man behöver namnge en variabel eller funktion för att kunna referera till den, men inte i sig är intresserad av namnet. Detta kallas *metavariabler* (eng. *metasyntactic variable*). I matematiken används namn som *x*, *y*, etc, men i datorsammanhang är namnen *foo*, *bar*, *baz* vanligare. Namnets ursprung är oklart, men kan ha att göra med amerikansk militärslangs förkortning FUBAR.

¹[GUI](http://en.wikipedia.org/wiki/Graphical_user_interface) (http://en.wikipedia.org/wiki/Graphical_user_interface)

B Vanliga tecken

Symb	Namn (<i>engelskt</i>)	Exempel på användning
#	Hash, brädgård (<i>hash, pound sign</i>). Används ofta som kommentartecken. I kombination med utropstecken, #! uttalas det <i>shebang</i> , se shellscript ◊ 7.1.3 .	
\$	Dollar (<i>dollar sign</i>). Miljövariabler i shellsript.	
*	Asterisk, stjärna (<i>asterisk, star</i>). Jokertecken ◊ 7.1.2 , används inom programmering, exempelvis <code>/* kommentarer i C */</code> .	
/	Snedstreck (<i>forward slash, solidus</i>). Begränsar kataloger i sökvägar ◊ 6.4.2 .	
\	Bakstreck (<i>backslash, reverse solidus</i>). "Escape:ar" (skyddar) tecken på kommandoraden. Begränsar kataloger i MS Windows.	
@	Snabel-a (<i>at sign</i>). E-postadresser; symboliserar mjuka länkar. Motsvarar svenska "à" (per styck) i engelskt språkbruk, "3 widgets @ 50¢ = \$1.50".	
()	parenteser (<i>parenthesis, brackets (GBR)</i>) Förutom normalt användande i text och matematik, också vanliga vid funktionsanrop i programmering: <code>printf("hello world\n")</code> .	
[]	hakparenteser (<i>square brackets, closed brackets</i>) Valfria argument i dokumentation; grupper av bokstäver på kommandoraden. Inom matematiken, ex <i>mängdlära</i> , $[a, b] = \{x \in \mathbb{R} a \leq x \leq b\}$	
{ }	klamrar, krullparenteser (<i>braces, curly brackets</i>) Begränsar block i många programspråk; expanderar till grupper på kommandoraden (<code>ls {alpha,beta,gamma}.txt</code>). Inom matematiken <i>mängder</i> (se exempel ovan).	
< >	Mindre än, större än (<i>less/greater than</i>). Används på kommandoraden för omdirigering (styra programs in/utmatning till filer eller andra program). Matematiska och logiska relationer. Används ofta för att markera text, istället för	
< >	vinkelparenteser (<i>angled brackets, chevrons</i>) Markera eller begränsa text. Blandas ofta ihop med mindre än och större än.	
« »	vinkelcitattecken; gåsögon (<i>guillemets, angle quotes</i>) Citerar (efter fransk förlaga) text, då vänd »baklänges« kring orden. » ska inte förväxlas med >> (dubbla större än), som används för omdirigering på kommandoraden.	
^	tak, circumflex (<i>caret, hat; circumflex (accent)</i>) Inom programmering, exempelvis för exponentiering (<code>5^3 = 5³</code>). Heter egentligen circumflex över bokstäver ("ê"), och caret annars.	
	vertikalstreck; pipe (<i>vertical line; pipe</i>) Kallas <i>pipe</i> när det används för att knyta ihop program på kommandoraden. Används inom matematiken för absolutbelopp, mängdlära, determinanter etc.	
~	tilde (<i>tilde</i>) Betecknar hemkatalog på kommandoraden. Spansk bokstav, används i vissa inlånade ord. Det dubbla tilde-tecknet, ≈ kallas i teckentabeller o dyl både "almost equal to" och "approximately equal to".	

Symb	Namn (<i>engelskt</i>)	Exempel på användning
'	apostrof (<i>apostrophe, (single) quote</i>). Citerar text. Det finns många typografiska tecken som kan förväxlas, exempelvis <i>'</i> (<i>prime</i>) som används inom matematik, och för att beteckna fot, bågminut, etc. Kallas ibland <i>"fnutt"</i> , när det är viktigt att skilja från citattecken.	
"	citattecken (<i>double quote; (straight) quotation mark</i>) Citerar text. Man skiljer på <i>typografiska</i> citattecken ("ord" på svenska, "word" på engelska) och <i>raka</i> ("ord") som också används för tum, bågsekund, etc. Kallas ibland <i>"dubbel-fnutt"</i> , när det är viktigt att skilja från apostrof. Typografisk konvention är för övrigt olika i svenska, amerikansk- och brittisk engelska:	<p>"Yttre text med 'inre text" "Outer text with 'inner text" 'Outer text with "inner text"</p> 
`	grav accent; <i>"bakfnutt"</i> (<i>grave accent; backtick, backquote</i>) Grav accent över bokstäver (" <i>è</i> "); används inom shellsript för att spara det ett program skriver ut på <i>stdout</i> .	
´	akut accent, accent (<i>acute accent</i>) Akut accent över bokstäver (" <i>é</i> "). Ej att förväxla med apostrof.	

C Program

Program	Kategori	Kommentar	Kommando
Internet			
Mozilla Firefox	Webbläsare	Standardwebbläsare på Ubuntu, finns även på andra system.	firefox
Mozilla Thunderbird	Epost	Standard för epost på CSCs Ubuntu.	thunderbird
SFTP	Filöverföring	Ubuntus filhanterare har inbyggt stöd för SFTP, om man går till <code>sftp://«användare»@dator.csc.kth.se/«sökväg»</code> . Men det finns även kommandoradsversion.	sftp
Office & dyl.			
OpenOffice	Office-svit	Gratis motsvarighet till MS Office och liknande program. Innehåller programmen <i>Writer</i> , <i>Calc</i> , <i>Impress</i> , mfl.	soffice
LaTeX	Typsättning	Skapar dokument utgående från en textfil med \LaTeX -kommandon. Hög inlärningsströskel, men användbart när man ska skriva mycket matematiska formler.	latex
LyX	Typsättning	En ordbehandlare som bygger på \LaTeX . (Är egentligen mer av ett grafiskt skal till \LaTeX .)	lyx
spell	Verktyg	Kommandorads-program för stavningskontroll. Finns också som kommando i Emacs.	spell
stava	Verktyg	Svensk motsvarighet till <code>spell</code> . Finns endast på CSC.	stava
Grafik, mediaapplikationer			
GIMP	Bildbehandling	Avancerat program för bildbehandling.	gimp
Inkscape	Vektorgrafik	Program för att redigera SVG, och i viss mån andra vektorformat.	inkscape
XFig	Vektorgrafik	Program för att skapa enklare vektorgrafik. Har verktyg för att arbeta ihop med \LaTeX	xfig
Rhythmbox	Mediaspelare	Ubuntus default-spelare.	rhythmbox

<i>Program</i>	<i>Kategori</i>	<i>Kommentar</i>	<i>Kommando</i>
GMplayer	Mediaspelare	Mediaspelare med tämligen minimalt gränssnitt. Känner igen de flesta mediaformat.	gmp1ayer
VLC	Mediaspelare	Mediaspelare som känner igen många mediaformat.	vlc
Texteditor			
Gedit		Ett (enligt programmets målsättning) enkelt och lättanvänt verktyg för att skriva ren text. Har stöd för plugins för mer avancerade behov.	gedit
Emacs		Kraftfull editor m m. Klarar av att redigera alla typer av text, och har fler verktyg än man kan räkna, och kan därför vara svårt att lära sig.	emacs
Programmering, Matematik			
Idle	Programmering	Enklare utvecklingsmiljö för Python-programmering.	idle [-n]
Eclipse	Programmering	Utvecklingsmiljö för Java	eclipse
Maple	Matematik	Program som kan räkna symboliskt (algebriskt) och numeriskt.	module add maple; maple
MATLAB	Matematik	Program för numeriska beräkningar. Har en inbyggd utvecklingsmiljö.	matlab
Subversion	Verktyg	Apache Subversion är ett generellt verktyg för versionshantering.	svn
Systemverktyg			
Appearance	Inställningar	Ändra utseende på skrivbord och fönster (bakgrund, tema, förvalda typsnitt, etc). System → Preferences → Appearance	gnome-appearan- ce-properties
Baobab	Kontoinformation	Visa hur mycket plats en katalog använder. Applications → Accessories → Disk Usage Analyzer	baobab

<i>Program</i>	<i>Kategori</i>	<i>Kommentar</i>	<i>Kommando</i>
CSC-verktyg			
Sima	Handledning	Kölista för hjälp på labbpas.	simamanager
Rapp	Kursresultat	System för att lagra kursresultat. Sköts av kursledaren, som efter avslutad kurs rapporterar in resultat i Ladok.	-
res	Kursresultat	Äldre system för att lagra kursresultat. Terminalbaserat, endast på Solaris.	res

Sakregister

- **, se asterisk
 - >*, *<*, *|*, se omdirigering
 - ~*, se tilde
- access-rättigheter, se AFS
- administratörer, se Systemgruppen
- AFS, **46**
 - access-rättigheter, **35**, **50**
 - kvot, **37**, **47**, **66**
 - rättigheter, **47**
 - UNIX-rättigheter eller, **35**
- allmänhandledning, **65**
- alt (tangent), se tangentbord
- ASCII, se teckenkodning
- asterisk, **73**
 - i fil-listning, **31**
 - som wildcard, **31**, **42**
- Autodesk Maya, **24**
- backup, se säkerhetskopior
- Baobab, **14**
- bash, se skal
- Chromium, **15**
- control (tangent), se tangentbord
- Creative Suite, **23**
- CSC, se även KTH
 - datorsalar, **5**
 - datortyper, **1**
 - regler, **2**
- datorkonto
 - CSC eller KTH, **66**
- datormus, **7**
- Disk Usage Analyzer, se Baobab
- dokumentation, **63**
 - online, **64**
- Dreamweaver, **24**
- Eclipse, se programming
- Emacs, **55–58**
 - tilde-filer, **57**
- encoding
 - iconv, **33**
- Evolution, **15**
- FAQ, **65**
- fillagring, **5**, se även AFS
- filnamnsexpansion, se tab completion
- filsystem, **39**, se även AFS
 - hemkatalog, **40**
- Final Cut Pro, **24**
- Firefox, **15**
- fjärrinloggning, se SSH
- gedit, **16**
- GIMP, **15**
- Gnome Desktop, **13**
 - skapa launcher, **14**
- GSSAPI, **53**
- GUI, Grafiskt användargränssnitt,
7, **72**
 - X11 forwarding, se SSH
- handledning, **64**
- hemkatalog, se filsystem
- hemsida, personlig, **50**
- iconv, se teckenkodning
- IDE, se programmering
- Illustrator, se Creative Suite
- InDesign, se Creative Suite
- Inkscape, **16**
- inloggning, **5**, se även Ubuntu, Mac OS X
 - felsäker, **12**, **25**
- inställningar
 - återställa, **67**
- ISO-8859-1 (Latin-1), se teckenkodning
- iWork (OS X), **22**
- Java, se programmering
- Java Desktop System, **26**
- jokertecken, se wildcard
- kataloger, se även filsystem
 - delade, **49**
 - rättigheter i, **50**
 - skapa och ta bort, **32**
- Kerberos, **6**, **46**
 - biljetter, **37**, **66**
 - hemifrån, **53**
 - kdestroy, **54**
 - kommandon, **47**
 - kpagsh, **53**
 - telnet, ftp, **51**
- Keychain (OS X), **21**
- kommandon, **30**
 - översikt & mnemonik för, **30**
 - a2ps, **33**
 - alias, **45**
 - apropos, **64**
 - awk, **45**
 - bg, **37**
 - cat, **33**
 - cd, **30**
 - chmod, **35**
 - clear-env, **67**
 - course, **50**, **69**
 - cp, **32**
 - diff, **45**

Sakregister

- echo, [37](#)
- emacs, *se* Emacs
- fg, [37](#)
- fs la, sa, [35](#), [49](#)
- fs lq, [47](#)
- fs mkm, [49](#)
- funktioner, *se* skal**
- grep, [45](#)
- iconv, [10](#)
- info, [63](#)
- kauth, kinit, klist, [47](#)
- kdestroy, [54](#)
- kill, [37](#)
- kpagsh, [53](#)
- less, [33](#)
- ln, [34](#)
- lpq, [33](#)
- lpr, [34](#)
- lprm, [34](#)
- ls, [31](#)
- man, *se* man-sidor
- mg, [58](#)
- mkdir, [32](#)
- module, *se* modules
- more, [33](#)
- mv, [32](#)
- nano, [58](#)
- pico, [58](#)
- pkview, [33](#)
- pts, [48](#)
- pwd, [30](#)
- res, [69](#)
- rm, [32](#)
- rmdir, [32](#)
- rsync, [32](#), [35](#)
- sed, [45](#)
- sima, *se* Sima Manager
- spell, *se* stavningskontroll
- ssh, sftp, scp, [14](#), [51](#)
- stava, *se* stavningskontroll
- svn, *se* Subversion
- tar, [32](#), [35](#)
- termer för, [38](#)**
- tokens, [47](#)
- tr, [45](#)
- väljare, [30](#)
- kommandorad, [28–41](#)**
 - detaljer, [42–54](#)
 - gränssnitt, [28](#)
- kpasswd, [29](#)
- KTH
 - regler, [2](#)
 - schema, [2](#)
- kurser
 - administrationsverktyg, [69](#)
- kvot
 - disk-, *se* AFS
 - utskrift-, *se* utskrifter
- länk, i filsystem, [34](#)
- lösenord, *se även* kpasswd
 - olika KTH-konton, [29](#)
 - problem att byta, [66](#)
- Λ_TE_X, [18](#)
 - Ly_X, [19](#)
 - T_EXShop, [23](#)
- logga ut, *se* utloggning
- märkspråk, [18](#)
- Mac OS X
 - inloggning, [20](#)
 - på CSC, [20–25](#)
- man-sidor, [63](#)
- markup language, *se* märkspråk
- matematik
 - skriva, [18](#)
- MATLAB, *se* programmering
- Maya, *se* Autodesk Maya
- meta (tangent), *se* tangentbord
- miljövariabler, [45](#)
- modifierare, *se* tangentbord
- moduler, [26](#), [46](#)
- MS Office, *se även* OpenOffice
 - Excel, [22](#)
 - för Mac OS X, [22](#)
 - Powerpoint, [23](#)
 - Word, [22](#)
- mus, *se* datormus
- Octave, *se* programmering
- OldFiles, *se* säkerhetskopior
- omdirigering, [33](#), [43](#)
- OpenOffice, [15](#)
- operativsystem, [1](#)
- OS X, *se* Mac OS X
- pausläge, [38](#)
- Photoshop, *se* Creative Suite
- pipeline, *se* omdirigering
- PostScript, [34](#)
- process, [37](#)
- program (som koncept), [37](#)
- programmering, [59–62](#)
- prompt, [29](#)
- public_html, *se* hemsida
- Python, *se* programming
- quota, *se* kvot
- rättigheter, *se* AFS
- redirect, [10](#)
- ren text, [16](#)
- säkerhetskopior, [48](#)
- sökväg, [29](#), [39](#)
- Safari, [21](#)
- script, [28](#), [43](#)
- shift (tangent), *se* tangentbord
- Sima Manager, [65](#)
- skal, [29](#)
 - fel, [67](#)

- funktioner, [45](#)
- Solaris, *se även* teckenkodning
 - inloggning, [25](#)
 - på CSC, [25–27](#)
- SSH, [14](#), [51](#), [67](#)
 - Kerberos och, [53](#)
- standard input/output/error, *se*
 - omdirigering
- stavningskontroll, [19](#)
- Subversion, [62](#)
- SunRay-terminaler, [25](#)
- support, *se* handledning
- suspend:a, *se* pausläge
- symbolisk länk, *se* länk
- Systemgruppen, KTH CSC, [65](#)

- tab completion, [29](#), [31](#), [43](#)
- tangentbord, [7](#)
 - compose (tangent), [8](#)
- tcsh, *se* skal
- teckenkodning, [10](#)
 - iconv, [10](#)
- Thunderbird, [15](#)
- tilde
 - filer, *se* Emacs
 - som sökväg, *se* filsystem
- tunna klienter, *se* SunRay-terminaler

- Ubuntu
 - historik, [1](#)
 - inloggning, [12](#)
 - på CSC, [12–19](#)
- UNIX, [1](#), [30](#)
- USB
 - Mac OS X, [21](#)
 - problem med, [66](#)
 - Ubuntu, [17](#)
- UTF-8, *se* teckenkodning
- utloggning, [13](#), [25](#)
 - forcera, [12](#), [25](#)
- utskrifter, [33](#)
 - kvot, [33](#)
 - Mac OS X, [24](#)
 - Ubuntu, [17](#)

- whitespace, [72](#)
 - citera, [38](#)
- wildcard, [31](#), [42](#)

- X11 forwarding, *se* SSH

D Lathund till UNIX-kommandon

D.1 Informationssökning, planering, andra användare

`apropos ord`
Sök efter manualsidor med nyckelordet `ord`.
`date`
Skriv ut aktuellt datum och tid.
`man ämne`
Visa manualsidan för ämne `ämne`.
`who`
Skriv ut vilka som är inloggade på denna dator.

D.2 Kommandon, processer

`bg`
Lägg en stoppad process i bakgrunden.
`cd`
Ändra aktuell katalog.
`clear`
Rensa terminalfönstret.
`echo arg`
Skriv ut `arg`, exempelvis `echo $PATH`.
`exit`
Avsluta ett shell.
`fg`
Lägg en stoppad process i förgrunden.
`jobs`
Lista egna aktiva processer.
`kill PID`
Avsluta process med "process identifier" `PID`, se `ps` nedan.
`passwd`
Ändra lösenord.
`execvera program`
Exekvera `program` i bakgrunden eller i fönstermiljö.
`ps`
Lista processer och deras `PID`ar, se `kill` ovan.
`ps -C program`
Ubuntu: Lista alla processer som heter `program`.
`ps -u användare`
Ubuntu: Lista alla processer som ägs av `användare`.
`ps -elf`
Ubuntu, Solaris: Lista alla (every) processer, lång och full lista.
`pwd`
Skriv vad som är aktuell katalog (se `cd` ovan).
`stty`
Avläs eller sätt terminalparametrar.
`ssh`
Terminalförbindelse mot annan dator.
`time kommando`
Tar tid på kommandot `kommando`.

D.3 Text och data i filer och pipes

`cat fill fill2`
Skriv ut `fill` och `fill2` som om de var en enda fil.
`cmp`
Jämför innehållet i två filer.
`comm`
Jämför rader i två sorterade filer.
`cut`
Väljer ut kolumner ur varje rad i en fil.
`diff`
Jämför innehållet i två textfiler.
`emacs`
Starta textredatorn GNU Emacs.
`grep text filnamn`
Sök efter strängen `text` i filen `filnamn`.
`head filnamn`
Skriv ut början av filen `filnamn`.
`less`
Textläsningsfilter som tillåter bläddring bakåt av texten.
`more filnamn`
Skriv ut filen `filnamn` med stopp vid varje skärmsida.

`paste`
Lägger samman rader ur olika filer.
`sort`
Sortera fil eller samsortera filer.
`tail filnamn`
Skriv ut slutet av filen `filnamn`.
`tee`
Filter som skickar sin utmatning både till en fil och till skärmen.
`tr`
Översätt (byt ut) tecken.
`uniq filnamn`
Rapportera lika rader i filen `filnamn`.
`wc filnamn`
Räkna tecken, rader och ord i filen `filnamn`.
`zcat`
Som `cat` fast för filer komprimerade med `compress` eller `gzip`.
`zmore`
Som `more` fast för filer komprimerade med `compress` eller `gzip`.

D.4 Filhantering

`chmod`
Ändra åtkomstkod för fil(er).
`fs [lals]`
Visa eller ändra AFS ACLer (*access control lists*).
`cp ofil nfil`
Kopiera `ofil` till `nfil`.
`cp fil katalog`
Kopiera filen `fil` till katalogen `katalog`.
`df`
Visa skivminnesutnyttjande.
`du`
Visa mängden skivminne som upptas av aktuell katalog och dess underkataloger.
`find file filnamn`
Söker efter filer genom att rekursivt gå igenom filkatalogen.
`file filnamn`
Undersök heuristiskt vilken typ filen `filnamn` har.
`gzip`
Komprimerar filer.
`gunzip`
Packar upp komprimerade filer (se även `tar`).
`ln -s`
Skapar en symbolisk länk till en annan fil.
`ls`
Lista namnen på filerna i den aktuella katalogen.
`ls -l`
Detaljerad listing av filerna i filkatalogen.
`mkdir katalog`
Skapa underkatalogen `katalog`.
`mv ofil nfil`
Byt namn på fil från `ofil` till `nfil`.
`mv fil katalog`
Flytta filen `fil` till katalogen `katalog`.
`rm filnamn`
Ta bort katalogen `filnamn`.
`rmdir katalog`
Ta bort katalogen `katalog`.
`sftp address`
Flyttar filer mellan lokal dator och datorn med `address` `address`.
`tar`
Hanterar arkiv av kataloger och filer.

D.5 Skrivhantering

`a2ps -pskrivare fil`
(Ubuntu) Skriv ut filen `fil` på skrivare `skrivare`.
`print -pskrivare fil`
(Solaris) Skriv ut filen `fil` (PostScript eller text) på skrivare `skrivare`.
`lpq -pskrivare`
Visa utskriftskön på skrivare `skrivare`.
`lprm -pskrivaren n`
Avbryt utskrift med jobbnummer `n` på skrivare `skrivare`.

E Lathund för Emacs

Escape kan användas som Meta. **ESC** **C-x** = **C-M-x**, etc. Även

E.1 Starta och lämna Emacs

emacs (i kommandotolken) starta Emacs
C-z textfält: stoppa Emacs temporärt
C-x C-c avsluta Emacs permanent

E.2 Filer

C-x C-f läs in en fil i Emacs
C-x C-s spara en fil på disk
C-x i sätt in innehållet i en annan fil i bufferten
C-x C-w spara som (ev. över existerande fil)

E.3 Att få hjälp

Tryck på **C-h** och följ instruktionerna. Om du är nybörjare, tryck **C-h t** för att få en demonstration. Alla hjälptexter är på engelska.

C-M-v bläddra i hjälpfönstret
C-x i ta bort hjälpfönstret
C-h a apropos; sök med hjälp av text
C-h c visa en tangents funktion
C-h f beskriv en funktion
C-h m få information om aktuell mode

E.4 Om något går fel

M-x recover-file rädda en fil efter en krasch
C-g avbryt delvis angivet kommando
C-x u eller **C-** återsätt till senaste ändring
M-x revert-buffer återgå till senaste sparade versionen
C-l rita om skärmen

E.5 Sökning

C-s sökning framåt
C-r sökning bakåt

Använd **C-s** eller **C-r** upprepade gånger för att upprepa sökningen eller upprepa den i någon riktning.

<Retur> avsluta sökningen
**** ångra senaste tangenttryckningen
C-g avbryt sökningen

Om Emacs fortfarande söker, kommer **C-g** avsluta den sökning Emacs just då håller på med. I annat fall kommer hela sökningen avbrytas.

E.6 Förflyttning

Förflyttning av markören

<vänster> eller **C-b** tecken bakåt
<höger> eller **C-f** tecken framåt
M-f ord framåt
M-b ord bakåt
<ned> eller **C-n** rad ned
<upp> eller **C-p** rad upp
C-a gå till radens början
C-e gå till radens slut
M-e mening framåt
M-a mening bakåt
M-< gå till början av bufferten
M-> gå till slutet av bufferten
C-v bläddra en skärmbild framåt
M-v bläddra en skärmbild bakåt
C-x < bläddra åt vänster
C-x > bläddra åt höger

E.7 Radering

C-d eller **** tecken framåt
<hak> tecken bakåt
M-d ord framåt
M- ord bakåt
C-k raden till slutet
M-k meningens till slutet
C-M-k syntaktisk enhet framåt
C-w radera aktiv region (klipp)
M-z tkn radera till nästa tecken tkn
C-y återställ raderad text (klisra)
M-y bläddra klisstrad text

E.8 Märkning

C-<spc> eller **C-@** sätt märket här
C-x C-x markera stycket
M-h markera sidan
C-x C-p markera funktionen
C-x h markera hela bufferten

E.9 Sök-och-ersätt

M-% interaktivt utbyte av text

Accepterade svar när man byter ut text interaktivt är

<spc> eller **y** byt den här, sök efter nästa

**** eller **n**

byt ej, sök efter nästa
byt denna, stå kvar
byt på alla ställen som återstår
gå tillbaka till föregående ställe
ändra utbytestext
avsluta

<Retur>

E.10 Flera vyer och fönster

En vy (eng. window) i Emacs är en del av dess fönster (eng. frame, programfönstret).

C-x 1 göm alla vyer, utom det aktiva.
C-x 0 göm denna vy
C-x 2 dela vyn i två delar, horisontellt
C-x 3 dela vyn i två delar, vertikalt
C-x 0 flytta markören till en annan vy
C-x 5 2 skapa nytt fönster
C-x 5 0 ta bort aktivt fönster

E.11 Formatering

<tab> indentera aktuell rad (beroende på mod)
C-M- indentera aktuell område (beroende på mod)
C-o sätt in radbrytning efter arbetspunkten
C-M-o flytta resten av raden vertikalt nedåt
C-x C-o radera blanka rader kring arbetspunkten
M-q radera alla blanka runt arbetspunkten
M-q jämna till stycket
M-g jämna till området
C-x f sätt högermarginal
C-x . sätt det prefix som varje rad börjar med

E.12 Ändra kast

M-u ändra ordet till VERSALER
M-l ändra ordet till gemena
M-c endast första bokstaven
ska vara versal
C-x C-u ändra området till versaler
C-x C-l ändra området till gemena
M-x capitalize-region som **M-c** men för ett helt område

E.13 Minibufferten

Føljande tangenter är definierade i minibufferten.

<tab> fyll i så mycket som möjligt
SPC fyll i maximalt ett ord
<retur> fyll i och utför
? visa möjliga alternativ

C-g avbryt

M-p, <upp> Føregående kommando i minibufferten

M-n, <ned> Nästa kommando i minibufferten

E.14 Bufferfar

C-x C-b lista alla bufferar
C-x b gå till en annan buffert
C-x k ta bort en buffert

E.15 Byt plats

C-t byt plats med föregående tecken
M-t byt plats med föregående ord
C-x C-t byt plats med föregående rad

E.16 Stavningskontroll

M-\$ kontrollera stavningen av ett ord
M-x spell-region kontrollera stavningen av området
M-x spell-buffer kontrollera stavningen i bufferten

E.17 Info

C-h 1 starta Info
SPC bläddra framåt inom nod
Del bläddra bakåt inom nod
. (punkt) gå till början av noden
n gå till nästa nod
p gå till föregående nod
u gå upp en två
? visa kommandon i Info
q avsluta Info

E.18 Tangentbordsmakron

C-x (starta inspelning
C-x) avsluta inspelning
C-x e utför senaste sparade makro
C-u C-x (utöka senaste makro
M-x name-last-kbd-macro döp senaste makro
M-x insert-kbd-macro visa listpok för makro i aktuell buffert