

Laborationer i kursmomentet Datoranvändning E1

Laboration nr 1: Vi inleder vår resa ut i datorvärlden

NB: Filer som omtalas finns tillgängliga vid första laborationstillfället

Originaltext: Olle Lundh

Omarbetning 95-08-21, 96-08-23, 97-08-20: Magnus Bondesson

Uppdatering 98-09-09: Thomas Hallgren

Syftet med denna första laboration vid datorn är framför allt att vi skall bekanta oss med den praktiska hanteringen av E:s arbetsstationer och förhoppningsvis uppleva dem som goda kamrater. Vi passar samtidigt på att lära oss använda ett par nyttiga program.

Symbolen markerar i uppgifter med flera delar inledningen på ett stycke med en konkret arbetsuppgift. Dessförinnan kan finnas motiverande eller förklarande text. Anmärkningarna (inleds med **Anm.**) kan genomgående överhoppas om du inte är specialintresserad.

Innan du kommer till laboration 1 skall du ha läst eller skummat följande avsnitt i Gula Boken: Introduktion (avsnitt 1.7-1.8.1, 1.12 och 1.17.4), Fönstersystemet X (kapitel 2). Det är inte lätt att läsa om sådant här om man inte har haft praktisk kontakt med UNIX-maskiner, så hoppa bara över det bland dessa sidor du tycker är konstigt och återvänd i stället efter laborationen.

Du behöver inte lämna in något från denna laboration. Men fyll för din egen skull ändå i svaren på frågorna på de markerade platserna.

Alla laborationer bygger på att du arbetar och provar dig fram självständigt. Ha Gula Boken till hjälp. Om du fastnar eller om det är något som du inte förstår så fråga handledaren. Sitt inte för länge och grunna på något du inte blir klok på. Texten i uppgifterna innehåller en del material från första föreläsningen, men det förutsättes nog att du deltagit i den.

Uppgift 1. Fönster och menyer



Logga in på gruppens konto för denna kurs och klicka bort det informationsfönster som visas. Du bör nu ha minst ett s k terminalfönster i vilket du kan skriva UNIX-kommandon. De flesta kommandon resulterar i att ett program startas och utför den begärda uppgiften. Du skriver kommandot och avslutar det med returtangenta (↵). När fönstret är redo att ta emot ett kommando skrivs det ut en kort ledtext, en s k prompt (sv redotecken), t ex

```
quar114:intro42:[~]> )
```

där `quar114` är datorns namn, `intro42` användarnamnet och placering i filsystemet anges inom klammrarna.

Innan du kan mata in text från tangentbordet till ett fönster måste fönstret göras aktivt. Det som du skriver på tangentbordet dirigeras vidare till det aktiva fönstret. Du

gör ett terminalfönster aktivt genom att flytta muspekaren in i fönstret. Ramen på fönstret ändrar då färg.

- Prova följande kommandon:

```
date  
more /u1/ext/hallgren/Intro/Berling.txt
```

I det andra fallet ser du en sida i taget av en längre välkänd (?) svensk text. Tryck på mellanslagstangenten för att övergå till ny sida. Notera också att du med hissen (inom vissa gränser) kan komma tillbaka till sådant som redan visats i fönstret.

- Undersök vad som händer då du med muspekaren i kombination med musknapparna pekar, klickar och drar på ett fönsters ram och de figurer som finns i ramens övre del. Du kan på detta sätt flytta fönster, ändra fönsterstorleken i sidled och/eller höjddled, stänga fönster etc. Prova de olika alternativen.

Nämner två sätt att flytta ett fönster: _____

Nämner två sätt att ändra storleken på ett fönster: _____

Hur ikonifierar/deikonifierar man ett fönster? _____

- Placera muspekaren på området utanför fönstren på skärmen, tryck sedan ned de olika musknapparna och titta på de menyer som dyker upp (dessa menyer kallas **bakgrundsmeny**). Vänster musknapp ger en meny där du bland annat kan logga ut (kommandot **Quit/Logout**; detta använder du alltså när ditt datorpass skall avslutas) och skifta mellan engelsk och svensk teckenuppsättning på tangentbordet (detta skall numera vara onödigt, eftersom det finns en kombinerad), mittenknappen ger en meny där du kan starta några olika typer av terminalfönster, högerknappen ger en meny där du kan starta några av de mest använda programmen. För att välja något i menyerna tar du fram aktuell meny och håller därefter musknappen nedtryckt samtidigt som du drar muspekaren på skärmen nedåt till önskat menyalternativ och släpper sedan musknappen. Om du inte redan har två terminalfönster, så skapa ett nytt med valet *Xterm* i mittknappsmenyn.

Terminalfönster (som kan startas via bakgrundsmeny enligt stycket ovan) emulerar (dvs fungerar som eller liknar) gammaldags textterminaler. Du kan ha flera terminalfönster igång samtidigt och arbete kan ske i dessa oberoende av varandra, dvs skenbart har du tillgång till många datorer samtidigt. Vi skall inte illustrera detta närmare just nu.

- Lägg fönstren så att de delvis överlappar. Hur kan man få det undre fönstret att hamna överst (det räcker att någon del av ramen är synlig)? _____

- Man kan i allmänhet kopiera åtminstone text mellan fönster. Markera genom dragning med musen (vänstra musknappen nedtryckt) ett kommando i det ena fönstret och klistra in det i det andra (mellersta musknappen; sker på den lilla svarta rutans plats).

Uppgift 2. Prova ett enkelt program

Starta med programmenyn (högerknappsmenyn) programmet *Calculator* och använd det. Det här programmet är ibland behändigt att ta till om man snabbt vill räkna ut

något. Sedan tvingas vi fundera på hur programmet avslutas. Moderna program har en egen menyrad med kommando för stängning men så är det inte här, varför vi får använda menyknappen längst upp till vänster för att stänga fönstret och därmed programmet. Det finns ett sätt till om man är utrustad med bra minne: placera muspekaren i fönstret och tryck på tangenten `q` (av quit).

Anm. Programmenyns *HP-Calculator* startar en variant. De båda versioner kan alternativt startas med UNIX-kommandot `xcalc -stipple -geo 300x400` respektive `xcalc -rpn -stipple -geo 400x250`. Det är alltså samma program, som startas med olika parametrar. Det finns också en speciell Digital-version som startas med `dxcalc`.

Uppgift 3. Prova ett annat enkelt program

- Starta med UNIX-kommandot `xpaint` ett ritprogram. Programmet är lättanvänt utan ingående instruktion och småröligt, men du kommer nog inte att ha så stor användning av det. Börja med att skapa en ritarea med **New Canvas** i **File**-menyn. Prova bl a kopiering/klistring med **Edit**-menyns kommandon. Programmet kan öppna och spara bilder, vilket sker på samma sätt som i kommande program, så strunta i det nu. Programmet har en egen menyrad och du avslutar det med **Exit** under **File**. Lagg märke till att programmet håller reda på att du gjort ändringar och ger dig en chans att spara innan det avbryts för gott. Bry dig dock inte om det just nu (filerna blir stora) utan besvara frågan med **No**.

Anm. Programmenyns *Paint* startar ett liknande men svagare program kallat *xpaint*.

- Starta samma program igen i avsikt att för kommande behov göra en liten bild. Skapa nu ritarean i stället med **New With Size** och välj storleken till 100 x 50. Rita bilden och spara den med **Save As** och under GIF-format som filen `MINBILD.gif`. Gå sedan ur med **Exit**. Skriv sedan i terminalfönstret `xv MINBILD.gif` för att kontrollera att den existerar (`xv` startar ett bildvisningsprogram). Bryt det genom att placera muspekaren i bilden och tryck på tangenten `q`.

Uppgift 4. Kommandotolken `tcsh`

När man startar ett terminalfönster startas ett program kallat kommandotolk (eng shell) som tolkar och utför de kommandon som du skriver. Den kommandotolk vi använder på E-linjens Unix-datorer heter *tcsh* (utläst T-C-SHell) och lagrar de kommandon som du ger till den så du behöver inte skriva jobbiga och långa kommandon mer än en gång. Du kommer åt de kommandon som du tidigare skrivit med hjälp av tangentbordets uppåtpil. Med nedåt- och uppåtpiltangenterna kan du bläddra fram och tillbaka bland de tidigare kommandona. Du kan också redigera kommandona med hjälp av höger- och vänsterpiltangenterna och på så sätt t ex. rätta kommandon som skrivits in felaktigt.

- Aktivera det första terminalfönstret och prova (börja t ex. med det felaktiga kommandot `datum` som sedan korrigeras till `date` utan omskrivning av de tre första bokstäverna). Verifiera också att `more`-kommandot från uppgift 1 finns kvar.

Uppgift 5. Filsystemet och några UNIX-kommandon

Program och datafiler ligger i UNIX ordnade i ett hierarkiskt filsystem. T ex har varje nybliven E1:a en egen katalog (eng. directory) kallad hemkatalogen i katalogen `/u1/e98`. Du kan alltid ta reda på vilken den aktuella katalogen är med kommandot `pwd` (print working directory). Vid inloggning hamnar man i hemkatalogen, dvs den aktuella katalogen är hemkatalogen.

- Vad heter gruppens hemkatalog? _____

Du kopierar filer med `cp`-kommandot (av copy) enligt

```
cp filnamn kopians_filnamn
```

- Skapa en kopia av filen `/u1/ext/hallgren/Intro/Berling.txt` hos dig och kalla den `Berling.txt` (OBS! du bör inte ha filnamn som innehåller mellanslag eller de svenska bokstäverna å, ä, ö, Å, Ä, Ö). Kontrollera därefter att kopian skapats med `ls` (av list files) eller `ls -l`. Kommandot `ls` i terminalfönstret skriver ut vilka filer som finns i den aktuella katalogen. Om du lägger till flaggan `-l` till kommandot `ls`, dvs skriver `ls -l` så får du lite mer information om respektive fil, bl.a. står när innehållet i varje fil senast ändrades, vem som äger filen och hur stor filen är.
- Tag nu en kopia av din egen fil och kalla kopian `Berling2.txt`. Kontrollera med `ls -l` igen. Byt sedan namn på denna andra kopia med `mv`-kommandot (av move) enligt `mv filnamn nytt_filnamn` till `Selma.txt`. Kontrollera igen med `ls -l` (använd piltangenterna!!!).
- Tag slutligen bort `Selma.txt` med `rm`-kommandot (av remove). Kontrollera.

Det är för framtida behov bra att veta att `..` syftar på katalogen ovanför den aktuella och att `.` avser den aktuella. Dessa beteckningar används bl a i vissa dialogrutor som vi snart möter. Tag med `ls -l ..` reda på vilka filer som finns i katalogen ovanför din.

Uppgift 6. "Surfa på nätet"

Man kan utföra väldigt mycket vid en dator utan att behöva bekymra sig om vanliga UNIX-kommandon. Det har vi redan sett i uppgifterna 2 och 3. Vi skall nu övergå till att titta litet på ett modernt kraftfullt program med hyggligt användargränssnitt. Vi väljer webbläsaren (eng. web browser) *Netscape*. Programmet i sig har ett betydande intresse, men från vår synpunkt sett är det väl så viktigt att det innehåller ingredienser som återkommer i andra kanske inte lika roliga program. Syftet är därmed inte att göra dig till en expert på nätverksvandring.

Anm. Beskrivningen gäller version 3 av webbläsaren. Version 4, var inte tillgänglig på E-sektionens datorer då detta skrevs. Den största konkurrenten *Internet Explorer* finns bara för vissa UNIX-varianter.

- Starta *Netscape* från programmenyn. Det dyker nog upp en sida med engelsk text. Klicka med musen någonstans på sidan och strax kommer du till E:s hemsida och kan fortsätta din vandring ut i världen. Det kan vara värt att notera att adressen (webbadressen) - eller referensen - till det dokument du betraktar anges i övre delen av fönstret med **Location**: `http://www.etek.chalmers.se/` (normalt anges ett

fullständigare namn, som i detta fall skulle avslutats med `index.html`). Detta betyder att det dokument som du ser på finns högst upp bland WWW-filerna på datorsystemet `etek.chalmers.se` och heter `index.html`.

- Minska höjden på fönstret och notera att du då får en hiss med vilken du kan glida över dokumentet. Prova och återgå sedan till ungefär samma storlek som från början.
- Ge dig nu ut på en vandring på egen hand. Notera att du med **Back**- och **Forward**-knapparna kan vandra fram och tillbaka mellan redan besökta sidor. Observera också att du med **Bookmark**-menyn kan spara adresser till intressanta sidor för snabbval vid senare tillfälle. Spara några adresser i den och testa. Med **Go**-menyn kan du hoppa mellan redan besökta sidor.

Uppgift 7. Fortsatt nätsurfning

Man kan också ta sig till ett godtyckligt dokument genom att i Location-rutan skriva in en adress (klicka först med musen där) och sedan trycka på returknappen (motsvarande kan också göras med knappen **Open** eller menyknappen **File/Open Location**).

- Tag dig på detta vis till t ex någon av adresserna

`http://www.aftonbladet.se`
`http://www.gp.se`
`http://www.nyteknik.se`

och se om du kan hitta någon god nyhet. Om du vid en dator som har ljud, tycker jag att du också bör passa på att besöka

`http://www.sr.se/ekot/nyheter/`

Välj EKOT och lyssna t ex på en del av senaste radioekot.

På adressen

`http://www.skolverket.se/skolnet/smultron/`

finns en utförlig introduktion till Internet och WWW. Klicka dig fram till avsnitt 2.9 och titta på listan med exempel på vad WWW används till idag.

Några andra adresser, som du nog inte hinner med nu, men som antyder rikedomen i WWW:

<code>http://www.lysator.liu.se/runeberg/</code>	Elektronisk litteratur
<code>http://www.sverigeturism.se/index.htm</code>	Turism
<code>http://www.passagen.se</code>	Telia marknadsplats
<code>http://www.torget.se</code>	Postens motsvarighet

De båda sista innehåller mängder med länkar, många inriktade mot en ung publik.

Som redan nämnts kan du spara intressanta adresser som s k bokmärken (eng bookmarks) med Bookmark-menyn.

Uppgift 8. Netscape som lokal informationshanterare.

Med webbläsare kommer man åt vissa filer på datorer runtom i världen, som annars skulle vara helt fördolda. Men man kan också använda en sådan för att presentera lokal (dvs på ett datorsystem) information på ett lättillgängligt sätt. Vi skall titta litet på det och samtidigt se en aning på hur en s k HTML-sida är uppbyggd.

- Använd kommandot **File/Open File** för att öppna filen

```
/u1/ext/hallgren/Intro/Berling.html.
```

Det går också bra att i Location-rutan skriva

```
file:/u1/ext/hallgren/Intro/Berling.html
```

och trycka på returtangenten. I det första fallet får du en dialogruta med vars hjälp du successivt tar dig fram till den önskade filen. Alla moderna program har sådana dialogrutor för öppnande av filer, men tyvärr varierar utseendet och beteendet en del. I det här fallet finns till vänster en kolumn med kataloger och till höger en kolumn med filer. Genom att i **Filter**-rutan byta ut den normalt avslutande * mot t ex *.html eller a*b kommer bara vissa av en katalogs filer att visas; i första fallet bara de vars namn avslutas med .html och det andra fallet bara de som inleds med a och avslutas med b. Asterisken * står alltså för en godtycklig följd av tecken. När man dubbelklickar på en katalog görs samma sak som vid enkelklick+Filter. Man väljer **OK** när det slutliga valet är gjort (man kan skriva in ett namn i **Selection**-rutan) direkt. Knappen **Cancel** tar du till om det var av misstag du började öppna. Dialogrutan kan förstöras vid behov precis som ett vanligt fönster. Hur som helst vad vi ser är en förskönad variant av den tidigare lagerlöfska texten.

- Öppna nu på samma sätt filen /u1/ext/hallgren/Intro/DemoHTML.html. Eftersom vi ser en blandning av bilder och text med olika utformning kan filen inte bara innehålla ren text utan också ytterligare anvisningar. Filen är fortfarande en textfil, så vi kan betrakta och bearbeta den med ett vanligt redigeringsprogram. Låt *Netscape* vara igång. Kopiera i ett terminalfönster filen till din egen katalog. Starta redigeringsprogrammet *NEdit* (från programmenyn). Öppna från det **din kopia** av filen. Detta görs med **File/Open** och via en dialogruta liknande Nescapes men utan någon **Selection**-ruta. Dubbelklick har ibland svårt att ta och då kan det vara bättre att trycka på **Filter**-knappen. Du ser nu den text som beskriver sidan och dess utseende. Studera den något. Jag har med kommentarer <!-- --> lagt in en del förklaringar. Ändra sidan i några avseenden och spara den under nytt namn i din katalog (ytterligare s k GIF-bilder finns i samma katalog som originalfilen). Öppna den sedan från *Netscape* och se om du lyckades göra rätt. Gör du ytterligare ändringar (och sparar dem!) kan du med **Reload**-knappen i *Netscape* direkt se vad de får för effekt.

- Jag tycker att du nu skall försöka få in din egen GIF-bild från uppgift 3, så att den syns på HTML-sidan.

Anm. Många bygger upp sina HTML-sidor på detta vis. Man borde dock använda en redigerare som är bättre anpassad till HTML; det finns en hel del sådana men ingen given segrare. Netscape Communicator 4 har en hyfsad sådan inbyggd.

Anm. Vill du göra egna HTML-sidor tillgängliga för hela världen måste du ta del av informationen på E:s hemsida. Se en kommande uppgift.

Uppgift 9. Sökning efter information i en kommersiell databas

Det finns ett växande antal kommersiella databaser som kan nås via WWW. En sådan är Britannica Online, som baseras på uppslagsverket Encyclopedia Britannica. Chalmers är abonnent på denna databas och du kan därför nå den utan lösenord el dyl från arbetsstationerna på Chalmers. Enklast är att gå via E:s hemsida (**Länkar - Britannica Online**).

För min generation är namnen Louis Armstrong (jazz-musiker) och Neil Armstrong (den första människan på månen) välkända och finns i alla uppslagsböcker. Det finns också en person med detta efternamn som haft stor betydelse för radiokommunikation och elektronik. Slarvigt och egenartat nog finns han dock inte nämnd i vår svenska Nationalencyklopedi, men ägnas åtskilliga rader i Encyclopedia Britannica. Tag med hjälp av Britannica Online, reda på

- Hans båda förnamn
- Vilken kommunikationsteknik uppfann och kämpade han för?
- Vad fanns dessförinnan (två bokstäver)?
- Hur dog han och vilket år?

Däremot letar man förgäves efter t ex barnboksförfattare som Enid Blyton, Edward Ellis och W.E.Johns men hittar Astrid Lindgren.

Uppgift 10. Leta efter böcker på biblioteket

Gå till CTH-bibliotekets hemsida:

`http://www.lib.chalmers.se`

Via den (det finns andra sätt) kan vi leta efter böcker (och även reservera böcker om man har lånekort). Användargränssnittet för detta är betydligt sämre än det vanliga, beroende på att programmet är utformat för datorer utan grafiskt användargränssnitt.

Välj **CHANS, bibliotekets katalog** på bibliotekssidan. Då startar ett program som visar upp ett fönster. Följ anvisningarna (ange användarnamnet CHANS först). Undersök t ex om det finns en bok av Richard Murray som handlar om robotteknik och huruvida den är utlånad eller inte. Naturligtvis kan du välja något annat namn av större intresse för dig.

Uppgift 11. WWW-kataloger och sökmaskiner (sök tjänster)

När man letar efter information finns det flera vägar att gå. Många personliga sidor innehåller utvalda länkar för ett eller flera intresseområden. Det ligger säkert en hel del möda bakom dessa sidor.

För seriöst arbete kan man ta sig till ett par ställen som innehåller strukturellt ordnade kataloger med adresser. Med knappen **Länkar** på E:s hemsida kommer man till en sida med länkar till bland annat ett antal kataloger över WWW-resurser. Exempel på

sådana är (inom parentes anges motsvarande direktadress) **SUNETs** katalog (<http://www.sunet.se/sweden/main-sv.html>; man hamnar i katalogen över svenska WWW-resurser; SUNETs hemsida är <http://www.sunet.se>) respektive **YAHOOs** internationella och svenska katalog (<http://www.yahoo.com> resp <http://www.yahoo.se>). Om du vill få en uppfattning om ökningen av registrerade sidor i Sverige kan du trycka på knappen **Nyheter** i SUNETs katalog.

Både SUNET och Yahoo har sökfunktioner som man kan använda istället för att bläddra sig fram i kategorierna för att hitta det man söker. En annan typ av söktjänst tillhandahålls av t ex AltaVista och HotBot (<http://www.altavista.telia.com> resp <http://www.hotbot.com>) som också dessa kan nås t ex via E:s länksida. Spara AltaVista-adressen som ett bokmärke efter du valt land och språk och tryckt på Go, så slipper du ifrån inställningarna senare.

SUNETs och Yahoos sökningar är mera begränsade än AltaVistas och HotBots, genom att de huvudsakligen söker i katalogen över registrerade sidor. AltaVista har samlat på sig en massa information om websidor i hela världen och söker alltså i denna information. Eftersom webben ständigt förändras är informationen aldrig helt aktuell, och de svar man får hänvisar därför ibland till websidor som är borttagna eller inte längre innehåller den sökta informationen.

Sättet att söka i de olika söktjänsterna är också litet olika. Om man skriver in många ord kan det t ex betyda att man för en lista över websidor där alla orden finns med, eller en lista över sidor där minst ett av orden finns med. Titta på hjälpsidorna för respektive sökmaskin för detaljer. Ofta kan man välja mellan enkel eller avancerad sökning.

- Tag dig på detta sätt till YAHOO och begrunda hur många programspråk det finns information om eller ge dig på något annat ämne av intresse.
- Besök även AltaVista eller HotBot och sök efter något av intresse.
- Besök svenska Yahoo eller SUNETs katalog över svenska WWW-resurser och se efter vad det finns för material om en svensk kommun som intresserar dig.

Uppgift 12. Enkät

- Det finns många användningsområden för WWW. Ett är att samla in information från användare, kunder etc. För att vi skall få se ett exempel på detta har jag gjort i ordning en rätt oskyldig enkät, som finns som HTML-filen (dvs öppna med **Open File** i Netscape) `file:/u1/ext/hallgren/Intro/enkat1.html`. Låt varje gruppmedlem besvara den. Eventuellt kommer vi i läsperiod 2 att säga något om vad som ligger bakom den teknik som används i detta sammanhang.

Uppgift 13. Pyssla mer med redigeringsprogrammet NEdit

- Om *NEdit* inte är igång så starta det igen. *NEdit* är nog det program som majoriteten kommer att använda för redigering av rena texter. Låt oss därför nu prova på några fler möjligheter. Öppna din fil `Berling.txt` (se uppgift 5). *NEdit* kan arbeta med flera filer samtidigt. Om du med **File/Close** stänger din tidigare fil, så kommer följande

File/Open att visa den nya filen i samma fönster som förut. Om du inte använder **File/Close** öppnas ett nytt fönster som lägger sig direkt ovanpå det gamla.

- Gå till en viss rad i filen. Leta efter någon textföljd (t ex präst eller sup) i filen. Skriv in egen text. Kopiera/klistra på vanligt X-sätt, dvs med vänstra/mellersta musknappen, eller med menykommando. Testa ångra-kommandot **Undo** som i detta program minns flera steg bakåt. Spara under nytt namn i din hemkatalog.

Uppgift 14. Avsluta Netscape och ett välment påpekande

Avsluta nu *Netscape* med **Exit** under **File** liksom *NEdit*. Avslutningsvis kan nämnas att mycket material kommer att finnas tillgängligt som HTML-dokument. T ex de lappar med *Frågor och Svar* som hänger i datorrummen (via knapp på E:s hemsida). Här avslöjas t ex var skrivare står och heter.

Du bör nog i fortsättningen lägga själva "surfandet" på nöjeskontot. Det är väldigt lätt att spilla mycket tid på en aktivitet som är rätt passiv till sin karaktär. Men jag hoppas att det framgått att en hel del nyttig information kan hämtas in den här vägen.

Uppgift 15. Byt konto

Vi skall i följande uppgift ägna oss åt något som man normalt gör från sitt personliga konto. För att inte båda gruppmedlemmarna skall behöva upprepa samma sak på sina personliga konton tar vi nu och byter till ett av de personliga kontona.

- Logga ut med **Quit/Logout** i vänster musknapps bakgrundsmeny (se uppgift 1) och sedan in på ett av de personliga kontona.

Uppgift 16. Vi gör en egen hemsida

Vi har tidigare tittat litet på hur en HTML-sida är uppbyggd och gjort någon sådan. Under förutsättning att motsvarande fil är läsbar för alla när andra användare på data-systemet HTML-sidan med `file:fullständig_sökväg`, men den är inte alls tillgänglig för utomstående personer.

Vi skall nu göra en HTML-sida som kan nås av hela världen med

```
http://www.etek.chalmers.se/~ditt_användarnamn
```

Av säkerhets skäl finns motsvarande fil nu inte bland de vanliga filerna, men du kan lätt flytta dig till och arbeta i den katalog där filen hamnar.

- Vi måste först skapa en katalog för sidan. Det gör du med kommandot

```
webfix
```

(som du alltså skriver i ett terminalfönster). Du får först reda på ett antal förhållningsregler när det gäller offentliga HTML-sidor. Läs dessa noggrant och följ dem framför allt i all framtid. Bland reglerna står också hur du verkligen skall förfara för att den önskade katalogen skall skapas. Gör så.

- Du har nu en personlig WWW-katalog,

```
~www/lib/students/e97/ditt_användarnamn
```

i vilken du kan arbeta och placera HTML-relaterat material. Kopiera den tidigare HTML-sidan (och eventuell GIF-bild) dit och kalla den `PROV.html`. Modifiera den eventuellt med *NEdits* hjälp. Kontrollera att du från *Netscape* kan nå den med

```
http://www.etek.chalmers.se/~ditt_användarnamn/PROV.html
```

Förmodligen är du inte fullt nöjd med denna sida som din hemsida, utan vill arbeta vidare med den på egen hand utanför laborationstid. När du är nöjd med den byter du namnet till `index.html`. Det medför två saker. Dels behöver inte filnamnet anges i adressen, utan alla kan använda det kortare skrivsätt som nämndes i inledningen. Dels kommer ditt verkliga namn under den följande natten att läggas in i en fil så att din hemsida kan nås från E:s hemsida!

Från din hemsida kan du referera till andra HTML-sidor och bilder i din WWW-katalog med enbart filnamnet. Ett par steg ovanför din katalog - i `~www/lib/` - finns bl a bilder som du kanske vill utnyttja. Referens görs då med `/filnamnet`.

Många är utomordentligt kreativa när det gäller webbsidor, men det finns naturligtvis också sidor som saknar såväl form som innehåll. En god inspirationskälla är sidor som andra förtjuser en med. Man kan nämligen i *Netscape* kika direkt på källkoden (**View/Document Source** eller i vissa fall **Frame Source**) och även spara sidor som HTML-dokument (som sedan kan redigeras med t ex *NEdit*). Till och med ingående bilder kan sparas (med hjälp av högra musknappen).

VILL DU VETA MER?

Via kursens hemsida

```
http://www.etek.chalmers.se/~hallgren/Eda/
```

kan du hitta länkar och annat material som har med föreläsningarna/laborationerna att göra. Nytt stoff läggs till successivt.

LITEN ORDLISTA

Logga in	- att identifiera sig för datorsystemet.
Logga ut	- att avsluta sitt arbetspass så att datorn är klar för nästa användare.
Terminalfönster	- program som emulerar (efterliknar) en gammaldags textterminal.
Returtangent	- ↵-tangenten på tangentbordet (=return).
Default	- det värde/den inställning som gäller då inget anges explicit.
WWW	- World Wide Web.
HTML	- Hyper Text Markup Language, det språk som beskriver de flesta av de dokument som Netscape och andra webbläsare gillar.
webb	- försvenskad form av WWW (jfr jobb av eng job).
webbsida	- allmän beteckning på ett dokument som läses via webben; kan vara en HTML-sida, bild, animering, eller ljudstycke.
hemsida	- den webbsida som en person eller en organisation presenterar sig med och som utgör ingången för ytterligare information; kan alternativt betyda den sida som webbläsaren startar.

Laborationer i kursmomentet Datoranvändning E1

Laboration nr 2: UNIX och e-post

NB: Filer o d som omtalas finns tillgängliga vid första laborationstillfället

Originaltext: Olle Lundh

Omarbetning 95-08-21, 96-09-05, 97-09-03: Magnus Bondesson

Omarbetning 98-11-23: Thomas Hallgren

Den här gången skall vi tränga in litet djupare i UNIX. Ytterligare UNIX-material kommer någon gång i framtiden. Dessutom lär vi oss att använda e-post (vid ett senare tillfälle kommer vi att titta på konferens/nyhetsystem). **Arbeta hela tiden på kurskontot.**

Innan du kommer till Laboration nr 2 skall du ha läst eller skummat följande sidor i Gula Boken (upplaga 1998): Litet om allt (1.9-1.14), Dokumentation (kapitel 3), Historiemekanismen (delar av 1.7.4) och Program och kommandon (kapitel 4, skumma bara). Hoppa över det bland dessa sidor du tycker är konstigt och återvänd i stället efter laborationen. Liksom förra gången återges en del föreläsningstoff.

Som tidigare markerar symbolen i uppgifter med flera delar inledningen på ett stycke med en konkret arbetsuppgift. Dessförinnan kan finnas motiverande eller förklarande text.

Kom också ihåg att laborationerna obligatoriska moment i kursen. Labhandledarna håller koll på vilka grupper som är klara med vilka labbar, så prata med en handledare när ni är klara med en laboration. Han kanske ställer några frågor för att kolla att ni har gjort uppgifterna (men man behöver kanske inte hinna med alla), så skriv ner svaren på de frågor som finns i anslutning till några av uppgifterna.

Om du inte hann med alla uppgifterna förra gången

Sörj inte över det. Någon är inte så viktig (Uppgift 16, att tillverka en hemsida), men gör gärna övriga uppgifter på egen hand utanför laborationstid (stöter du på problem får du gärna fråga vid laborationstillfället).

Uppgift 1. Enkäten

- Om du inte redan besvarat enkäten (Uppgift 12 på Laboration nr 1), gör det nu.

Uppgift 2. Kursens hemsida

- Om du inte redan har besökt kursens hemsida, gör det nu. Som nämndes i slutet av Laboration nr 1 är adressen

<http://www.etek.chalmers.se/~hallgren/Eda/>

Sätt ett bokmärke så att du lätt kan komma tillbaka till sidan.

- Följ länken till E-sektionen. Om man där följer länken Datorhjälp hittar man svar på diverse frågor om datorsystemet på E. Detta kan också vara en bra sida att ha ett bokmärke på.

Vilka program som finns på E-sektionen kan man gratis ta hem och köra på sin egen dator? _____

Uppgift 3. Elektronisk post (eng. e-mail, sv. e-post)

Detta handlar om att kunna skicka meddelanden per dator till en eller flera personer och kunna läsa sådana som andra personer skickat till dig.

I UNIX-system har alla användare normalt en egen brevlåda. Brevlådans adress är samma som kontonamnet, t ex `introxx` för de konton som används i denna kurs. Är UNIX-system anslutet till Internet kan man, förutom att skicka meddelanden till andra användare på samma system, även skicka meddelanden till och ta emot meddelanden från användare på andra Internet-anslutna datorsystem. När man skickar meddelanden mellan olika datorsystem är adressernas utseende *användare@organisation*, där organisation normalt består av flera delar med punkt mellan. Tecknet @ uttalas som det engelska ordet *at* (ibland hör man också folk säga *snabel-a* eller *kanelbulle*). E-teknologer på Chalmers tillhör organisationen `etec.chalmers.se`.

Det finns flera program som man kan använda för att skicka och ta emot e-post. Vi tar en titt på hur e-post fungerar i *Netscape*.

Innan vi börjar skicka e-post-meddelanden utför vi en engångsåtgärd. Man brukar ju skriva under brev man skickar med sitt namn (och kanske något mer). För att slippa skriva under varje gång man skickar ett nytt brev kan man skapa en fil med namnet `.signature` i sin hemkatalog. Många e-post-program lägger då automatiskt till innehållet i denna sist i de brev man skickar.

- Skapa med *NEdit* i hemkatalogen `.signature` som innehåller era namn och labgruppens e-postadress.
- Starta nu *Netscape* om programmet inte redan är igång. Om du inte redan lagt märke till det så uppmärksamma att man får hjälptexter om markören får vila en stund på en ikon. Välj menyalternativet **Window/Netscape Mail**, så dyker det upp ett postfönster (första gången visar sig först en dialogruta om att katalogen `nsmail` inte existerar; klicka bara på **OK**). Klicka på bilden **Inbox**, så får du se ett välkomstbrev från Netscape, men några andra brev finns nog inte.

Längst upp till höger på bildskärmen finns en postlåda. På den kan man se om det finns nya meddelanden. Även Netscape undersöker med jämna mellanrum om det finns nya meddelanden i brevlådan och ändrar utseende på indikatorn i nedre högra hörnet av fönstret. Om Mail-fönstret är ikonifierat ändrar ikonerna utseende.



- För att läsa nyinkomna meddelanden ska man trycka på knappen **Get Mail**. Gör det. Förhoppningsvis dyker ett meddelande från Thomas Hallgren upp i **Inbox**. Läs det. Antagligen vill Thomas att ni skickar ett svar.
- Nu skall vi svara på brevet. Tryck på knappen **Re: Mail** (Re som i reply). Vi får nu upp ett nytt brevskrivfönster, där adressaten (Mail To:) och rubriken (Subject:) redan är

ifyllda. Vidare inkluderas det ursprungliga brevet text, vilket brukar vara bra när det gäller frågesammanhang, men du kan naturligtvis redigera bort delar. Skriv ditt svar (du kan redigera med musens hjälp, etc, ungefär som i *NEdit*). Skicka iväg meddelandet genom att trycka på knappen **Send**.

- Skicka e-post till den egna gruppen. För att få fram ett fönster att skriva brevet, klicka på knappen **To: Mail**. Fyll i adressen och rubriken (eng. subject) och skriv brevet på ett naturligt sätt. Om du vill skicka en kopia till någon kan du fylla i en adress till vid **Mail CC** (CC=Carbon Copy). Kontroll-läs brevet och tryck sedan bara på **Send**-knappen.
- Nu skall vi strax läsa brevet. När du skickat iväg brevet så vänta en stund till dess postlådan i övre högra hörnet av skärmen indikerar att du fått post (den reagerar dock långsamt så du kanske inte orkar vänta så länge). Tryck på knappen **Get Mail** igen, och notera att brevet dyker upp i **Inbox**. Notera också att du med musens hjälp kan vandra runt bland de brev du nu har i **Inbox**. En kopia av skickade brev hamnar för övrigt alltid i mappen **Sent**, som du också kan inspektera. Återvänd till det mottagna brevet i mappen **Inbox**.
- Du kan ta bort ett brev, genom att först markera det i förteckningen över brev och sedan trycka på **Delete**-knappen (i menyn!). Brevet flyttas då till mappen **Trash** (sv. soptunna, papperskorg). Där ligger det kvar till dess du använder menyalternativet **File/Empty Trash Folder**.
- Skicka eventuellt e-post till någon som sitter vid en dator intill dig.
- Man kan till och med skicka brev med bilder (och ljud etc). Detta förutsätter dock att mottagaren använder *Netscape* (eller liknande) för postläsning, så det skall man nog inte använda okritiskt. För detta används **Attach**-knappen, som låter oss i en dialog ange de filer (GIF m fl), som skall ingå. Prova - om du hinner under labpasset - detta. Skicka brev till den egna gruppen med någon GIF-bild (om inte annat kan du hitta sådana i `~hallgren/Intro/`). Läs brevet.
- Allra sist stänger du **Mail**-fönstret med **Close** och avslutar *Netscape* med **Exit**. (Om man inte avslutar *Netscape* innan man loggar ut kommer *Netscape* antagligen att klaga nästa gång det startas.)

Uppgift 4. Andra program för elektronisk post

Netscape är ett stort program som kan ta lång tid att starta. Det finns dock andra e-post-program man kan använda, om man t ex bara snabbt vill kontrollera den senast anlända posten, eller skicka ett litet enkelt meddelande. Exempel på sådana är `elm`, `pine` och `mail`. De två förra beskrivs i Gula Boken. Gemensamt för alla tre är att de körs i ett terminalfönster. Detta gör att de kan passa bra att använda när man är inloggad på E-sektionens datorer via modem.

Varning för Netscape

Många program för e-post tillåter att man sorterar sin post i olika mappar. I de flesta program ligger meddelanden kvar i användarens brevlåda tills de flyttas till en annan map. *Netscape* skiljer sig dock på denna punkt: när man trycker på knappen **Get Mail**

flyttas genast alla nyinkomna meddelanden bort från brevlådan till mappen som heter **Inbox**. Detta kan leda till två problem:

- När man har läst sin post med Netscape syns inte meddelandena längre i andra e-post-program.
- Om inte det lediga filutrymmet i användarens hemkatalog är tillräckligt stort för att lagra nyinkomna meddelanden, så kan man inte läsa dem med *Netscape*.¹

I nästa uppgift tittar vi på hur man kan hålla koll på det lediga filutrymmet och undvika det senare av dessa två problem.

Uppgift 5. Filsystemet: hur vet man hur mycket ledigt utrymme det finns?

Även om hårddiskar blir större och billigare i ganska snabb takt är lagringskapaciteten hos dagens datorer begränsad. Att filsystemet är fullt är ett fel som inte alla programmerare tänker så mycket på när de skriver program. Ett välskrivet program kanske säger "filen kan inte sparas eftersom det inte finns tillräckligt med ledigt utrymme" då detta fel uppstår. Ett mindre bra program kanske säger "write error" och något felnummer. Ett ännu sämre program säger kanske ingenting utan sparar halva filen och fortsätter sedan som om inget hade hänt.

Det finns alltså anledning att hålla lite koll på hur mycket ledigt filutrymme man har. Kommandot `quota` ger information om hur mycket filutrymme den påloggade användaren får lov att använda.

```
hortensia:hallgren:[~]> quota
Disk quotas for user hallgren (uid 1503):
  Filesystem blocks  quota limit  grace   files quota limit  grace
  /ul/ext      468  25000 30000    62    0    0
```

I exemplet ovan frågade användaren `hallgren` efter sin "quota". Kolumnen `blocks` visar hur mycket utrymme som redan används (1 block = 1 Kbyte på våra datorer), kolumnen `quota` visar den "mjuka" gränsen för hur mycket man får ha. Man kan tillfälligt överskrida den mjuka gränsen upp till den "hårda" gränsen, som visas i `limit`-kolumnen, men efter en viss tid (som anges av `grace`-kolumnen) kommer då som straff den hårda gränsen att sänkas till precis så mycket man har för tillfälligt, och man kan inte lagra en enda byte till. Man blir då tvungen att ta bort filer tills man kommer under den mjuka gränsen.

Det finns även en gräns för hur många filer man får ha. De fyra sista kolumnerna (`files`, `quota`, `limit`, `grace`) visar hur läget är med antalet filer, men fungerar annars på samma sätt som de fyra föregående kolumnerna.

Hur mycket filutrymme får er labgrupp använda? _____

Man borde få en varning (t ex varje gång man loggar in) om man har överskridit den mjuka gränsen, men det verkar inte som man får det på E-sektionens datorer. Ta för vana att kolla läget med `quota`-kommandot lite då och då.

Det finns ett kommando till som ger information om hur mycket filutrymme det finns:

1. Jag vet inte hur bra felkontroll Netscape har i denna situation. Om det finns brister här finns alltså risken att nyinkomna meddelanden försvinner...

```
hortensia:hallgren:[~]> df .  
Filesystem      1024-blocks      Used   Available Capacity  Mounted on  
/u1/ext@joakim  12331440    1765020    367864    83%    /u1/ext
```

I exemplet ovan undersöker användaren `hallgren` hur mycket plats det finns på den hårddisk (eller, rättare sagt, det filsystem) hans hemkatalog lagras på. Det kan faktiskt hända att hårddisken blir full utan att man har nått sin quota-gräns (t ex om många andra användare samtidigt sparar stora filer).

Anm. Punkten i kommandot `df .` anger att man bara vill se information om den aktuella katalogen (mer om detta senare). Utelämnas den visas information om alla tillgängliga filsystem.

- Hur stor är den disk som gruppens hemkatalog lagras på? Hur mycket ledig plats finns det? _____

Uppgift 6. Nyhetsgrupper

Nyhetsgrupper (även kallade diskussionsgrupper, eng Usenet news) är ytterligare ett sätt att sprida information via internet. Om du missade föreläsningen då jag pratade om detta, ta en titt på webbsidan

<http://www.skolverket.se/skolnet/smultron/news.htm>

som ger en kort introduktion och länkar till mer information.

Att läsa/skriva meddelanden i nyhetsgrupper görs på nästan samma som man läser/skriver e-post-meddelanden. Precis som det finns speciella program för e-post, finns det också speciella program för nyheter. *Netscape* går också bra att använda: menyalternativet **Windows/Netscape News** öppnar *Netscapes* nyhetsfönster.

- Öppna nyhetsfönstret i Netscape och notera att det dyker upp några nyhetergrupper uppe till vänster. (det kanske är för trångt från början för att man ska se vad grupperna heter, men man kan dra i kolumnrubrikerna för att skapa mera plats. Fråga handledaren om du inte kommer på hur man gör.)
- Det borde dyka upp en grupp som heter **news.announce.newusers**. Välj den och läs i meddelandet "Emily Postnews Answers Your Questions on Netiquette". Vad ska man göra, enligt Emily Postnews, om man har skickat ett meddelande och glömt skriva under?

Notera också att det finns ett meddelande med rubriken "A Primer on How to Work With the Usenet Community". Den kan vara bra att läsa innan man börjar skicka meddelanden till nyhetsgrupper.

I nyhetsfönstret syns normalt bara de nyhetsgrupper som man prenumererar på och som innehåller olästa meddelanden. I **Options**-menyn kan man ändra på det. Alternativet heter **Show ... Newsgroups**, där ... är **Subscribed**, **Active** eller **All**. Välj alternativet **Subscribed**. Vilka grupper relaterade till E-sektionen visas? _____

- Läs meddelandena i de grupper som har med E1 att göra. Kanske hittar du ett meddelande med ytterligare någon uppgift från Thomas Hallgren. Gör även denna uppgift.

Uppgift 7. Filsystemet: kataloger

Det är bra med ordning och reda i de flesta sammanhang. Eftersom du säkert på ditt personliga konto senare kommer att lägga mycket annat material, kan det vara bra ha särskilda *kataloger* (på engelska *directory*) för bl a olika kurser, t ex en katalog som har att göra med ditt projekt och en annan för matematik, osv. Kataloger skapas du med kommandot `mkdir` och du tar bort tomma kataloger med kommandot `rmdir`. I Windows 95 liksom MacOS använder man ordet *mapp* (på engelska *folder*) i stället för katalog och i Gula Boken kallas de bibliotek.

- Skapa en katalog *Datorintro* och flytta någon av gruppens filer dit. Flyttning gör vi ju med kommandot `mv källa destination` (se Uppgift 5 i Laboration nr 1). Källan skall vara ett filnamn och hittills har även destinationen varit det. Men den kan också vara en katalog, varvid det ursprungliga namnet bevaras. Hur gjorde du?

- Med kommandot `pwd` (print working directory) kan du se var du befinner dig i filträdet (redotecknen visar det också). Kommandot `cd katalognamn` använder du för att förflytta dig ned i katalogen med namnet katalognamn. Flytta dig till *Datorintro* och kontrollera på sedvanligt sätt att filerna finns där. Kommandot `cd ..` (`cd`, mellanlag, två punkter) använder du för att gå ett steg upp i fil-trädet. Du kan alltid ta dig tillbaka till kontots hemkatalog med kommandot `cd` (utan katalognamn). Prova bägge sätten att från *Datorintro* återvända till hemkatalogen.

Uppgift 8. Filsystemet: Kopiering

Det händer att man vill kopiera mellan olika användare, t ex från ett kurskonto till ett personligt konto. Man ser då till att vara inloggad på det konto till vilket kopiering skall ske. Kopieringen måste gälla en fil som är läsbar (se nästa uppgift) för alla.

Med `~användarnamn` avser man hemkatalogen för en viss användare, vilket oftast är bekvämare än att skriva ut fullständigare namn. Tecknet `~` utläses "tilde". Tex avser `~intro130` och `/u1/kurs/intro/intro130` samma katalog och `~e98xxxx` samma som `/u1/e98/e8xxxx`.

Med `.` avser man aktuell katalog (`..` som vi mött tidigare avser katalogen ovanför).

Här ser du några varianter på en kopiering som du gjort tidigare:

```
cp ~hallgren/Intro/Berling.txt .
cp ~hallgren/Intro/Berling.txt Berling.txt
cp /u1/ext/hallgren/Intro/Berling.txt ~introxx
cp ~hallgren/Intro/Berling.txt B.txt
```

De tre första innebär samma sak (om den aktuella katalogen är gruppens hemkatalog), medan i den sista kopian får ett annat namn.

- Kopiera en fil från någon av gruppmedlemmarnas personliga konton eller i nödfall från *hallgren* till katalogen *Datorintro* och som skapades i Uppgift 7. Kontrollera.

Uppgift 9. Filrättigheter

När du gjort i ordning en fil vore det i allmänhet inte så bra att vem som helst kunde ändra i den (inklusive ta bort den). Normalt har därför bara användaren (eng. user), dvs du själv, ändringsrätt - kallad **skrivrätt** - när det gäller filer och kataloger. Däremot har det i UNIX-världen, åtminstone bland akademiker, varit vanligt att man låter andra ha **läsrätt** till många filer. E-sektionens datorsystem har valt denna form av öppenhet, vilket betyder att filer normalt kan läsas av andra användare.

Kommandot `ls -l`, som vi sett förut, visar en massa information om filerna i en katalog, bland ann vilka rättigheter som gäller. Avsnitt 1.13 i Gula Boken beskriver detta.

Vi låter nu hallgren ta reda på information om alla filer i några kataloger (* betyder noll eller flera godtyckliga andra tecken, dvs KATA* står för alla namn som inleds med KATA):

```
hortensia:hallgren:[~]> ls -l Intro Intro/KATA*
Intro:
-rw-r--r--  1 hallgren users          4491 Aug 19 17:42 DemoHTML.html
drwxr-xr-x  2 hallgren users          8192 Sep  2 16:24 KATALOG1
drwx-----  2 hallgren users          8192 Sep  2 16:24 KATALOG2
-rw-r--r--  1 hallgren users          3547 Aug 25 15:12 enkat1.html

Intro/KATALOG1:
total 2
-rw-r--r--  1 hallgren users           4 Sep  4 15:56 fil1
-rw-----  1 hallgren users          13 Sep  4 15:59 fil3
Intro/KATALOG2:
total 1
-rw-r--r--  1 hallgren users           5 Sep  4 15:57 fil2
```

Filen `fil1` är läsbar för alla, medan `fil3` bara är läsbar för ägaren. Filen `fil2` är trots att det finns `r` rakt igenom inte läsbar för andra än ägaren, eftersom katalogen `KATALOG2` som innehåller filen inte är det.

- Försök att kopiera var och en av filerna `fil1`, `fil2` och `fil3` enligt ovan till din egen katalog. Katalogerna som innehåller filerna finns i katalogen `~hallgren/Intro/`. Förklara för dig själv.
- Flytta dig med `cd` till `KATALOG1` och gör `ls -l`. Flytta dig med `cd` till `KATALOG2`. Förklara för dig själv.

Nu uppstår frågan hur man ändrar rättigheterna, ty ibland vill man ju att andra inte skall ha läsrätt till ens filer. Man använder kommandot `chmod` (change mode) för detta. Text betyder `chmod go-rw minfil` att filen `minfil` inte kan läsas eller skrivas av gruppen eller andra (+ inför rättighet). I enlighet med E-sektionens filosofi om öppenhet bör nog inte hemkatalogens rättigheter ändras, utan bara de för underliggande filer och kataloger. Vissa program, t ex `postprogram`, skapar kataloger som är lässkyddade från början.

- Ändra nu rättigheterna för någon av filerna i gruppens hemkatalog. Kontrollera med `ls -l` resultatet.

Uppgift 10. Kommandotolken *tosh* igen

Kommandotolken *tosh* är ännu listigare än vad vi såg i Uppgift 4 i Laboration nr 1.

Programmet *tosh* har viss förståelse för hur kommandon och filnamn som är vanliga kommandoargument ser ut. Om du trycker på <TAB> när du börjat skriva ett långt kommando eller filnamn kommer *tosh* att försöka fylla i resten av namnet åt dig. *tosh* kommer att fylla i så många bokstäver framåt som kan matchas unikt mot tillgängliga kommandon och filnamn. Om t ex flera filnamn börjar på samma sätt måste du hjälpa till genom att fylla i tecken som skiljer dem åt. Du kan därefter trycka på <TAB> igen för att få resten av namnet ifyllt.

- Prova med att t ex skriva `da` i ett terminalfönster och tryck därefter på <TAB> så fyller *tosh* i resten. Vilket kommando blev det? _____

- <Ctrl-D> kan användas för att se de alternativ som återstår när du skrivit delar av t ex ett kommando eller en sökväg i systemets filhierarki. Skriv ett `x` på en tom rad och håll därefter CTRL-tangenten nedtryckt samtidigt som du trycker på D-tangenten (= <Ctrl-D>). Programmet *tosh* listar då samtliga program i sökvägen (sökvägen talar om i vilka kataloger som *tosh* skall leta efter program) som börjar på bokstaven `x`. Fortsätt med att efter `x`:et skriva ett `c` och tryck sedan <Ctrl-D>. Du får nu en lista på de program i sökvägen som börjar på `xc`. Fortsätt med ett `a` (dvs `xca`) och <Ctrl-D>. Som du ser finns det bara ett program som börjar med `xca`. För att skriva ut hela namnet behöver du nu bara trycka på <TAB> så fyller *tosh* i det sista. Starta sedan programmet genom att trycka på Return-tangenten. Hur många program börjar på bokstäverna `xc`? _____

Uppgift 11. Söka information om UNIX-kommandon

Moderna program (som *Netscape*, *NEdit* och *FrameMaker*) har inbyggda hjälpsystem som är rätt vänliga mot användaren. När det gäller information om UNIX-kommandon är den tekniskt fullödlig, men den vänder sig knappast till novisen och man blir lätt skrämmd när man överöses med alla detaljer och finesser. Vet man vad kommandot heter så kan man använda `man kommandonamn` för att få fram ett sk manualblad (ofta flersidigt!). Känner man inte till kommandot kan man använda `apropos ledord`, som listar ett antal kommandonamn - som kan tänkas ha med ledordet att göra - med kort information om varje. Man kan sedan använda `man`-kommandot för att få ytterligare information. Med *Manual* i programmenyn kan man också leta sig fram till ett kommando och dess manualblad.

- Titta på manualbladen för `cd`, `mkdir` och `chmod`.
- Använd `apropos` i syfte att hitta t ex redigeringsprogram, pascalkompilatorer, fortran-kompilatorer, skrivarkommandon eller C-kompilatorer. Lämpliga ledord kan vara `editor`, `pascal`, `print`, `fortran` respektive `c` (varför är resultatet av `apropos` i det sista fallet att se som ett misslyckande?).
- Sök reda på tre program som visar vad klockan är (prova programmen när du sökt rätt på dem):.....

Det program av de tre ovan vars namn börjar med `xc` visar i normala fall bara en timvisare och en minutvisare. Hur får man programmet att även visa en sekundvisare (använd man-kommandot och lusläs bladet)?

Ytterligare ett sätt att få fram information om systemet är att använda *Bookreader* som du hittar i Program-menyn. Med den kan man läsa manualer för Digital's systemprodukter, t ex användarmanualer för pascal- och C-kompilatorna. Vi struntar i det.

Uppgift 12. Förgrunds- och bakgrundsjobb

Som bekant kan man ha ett program gående per terminalfönster. Program som startas från bakgrundsmenyn lägger i allmänhet inte beslag på något terminalfönster. I själva verket kan man arbeta med flera program i ett fönster. Om bl a detta handlar denna uppgift.

- Skriv `xclock -update 1` i ett terminalfönster och tryck därefter på return. En klocka startar.
- Stanna programmet `xclock` genom att ge kommandot `<Ctrl-Z>` i terminalfönstret. Observera att du inte avslutat programmet utan bara stannat det.
- Lista det program som du tillfälligt stannat med kommandot `jobs` och starta programmet igen med `fg %1` (eller bara `fg` om `jobs` bara listar detta enda program).
- Stanna programmet igen och starta sedan det i bakgrunden med `bg %1`. Du har nu fått tillbaka redotecknet (promptern) i terminalfönstret trots att programmet `xclock` är igång. Du kan nu starta nya program i terminalfönstret.
- Starta `xclock -update 1` en gång till men den här gången skall du starta programmet i bakgrunden direkt. Hur gör du (se efter i Gula Boken)? _____
- Lägg en av klockorna i förgrunden och avsluta sedan programmet genom att ge kommandot `<Ctrl-C>`. Den andra klockan avslutar vi i nästa uppgift.

Uppgift 13. ps och kill

Det händer någon enstaka gång att ett program låser sig. Då kan man ha nytta av det följande.

Med kommandot `ps ux` kan du få en lista på vilka processer du har. I listan kan du se att varje process har ett unikt nummer som kallas PID (process identification number) kopplat till sig. Med kommandot `kill` och med PID som argument kan du skicka signaler till processer. Du kan t ex ta död på en process med `kill PID`.

- Tag död på klockan från förra uppgiften med `kill`. Du kan givetvis bara ta död på dina egna processer.

Uppgift 14. Extra-uppgift: Programmering och kompilering

Den här kursen skall inte handla om programmering i något traditionellt programspråk, men kanske kan det ändå vara kul att prova på litet grand.

- Filen `~hallgren/Intro/FirstProg.c` innehåller källkoden till ett korrekt program i programspråket C. Kopiera den till din egen katalog. Kompilera den med

kompileringskommandot `gcc` eller `cc`. Kompilatorn lämnar ifrån sig den exekverbara filen `a.out`. Provkör programmet.

- Programmet beräknar summan av alla heltal från 1 till 1000 genom brutal räkning. Redigera programmet med *NEdit* så att summan av de 1000000 första positiva heltalen beräknas i stället. Kompilera på nytt och kör det.

Laborationer i kursmomentet Datoranvändning E1

Laboration nr 3: Matematikverktyget Maple

1992-11-05, 93-11-01, 94-10-31, 95-10-09, 96-09-13, 97-09-10 Magnus Bondesson

98-09-22 Thomas Hallgren

1 Syfte

Innan du kommer till laborationen skall du som vanligt ha tittat på några sidor i Gula Boken, denna gång kapitel 9 om Maple, fram till och med avsnitt 9.7 (hoppa över 9.3). Läs också åtminstone avsnitten 1-2 i detta laborations-PM.

Syftet med denna laboration är att ge dig en första praktisk kontakt med matematikverktyget Maple. Vissa av uppgifterna avser dessutom att antyda hur dessa verktyg kan användas för att ge ökad förståelse kring något fenomen. En del uppgifter kanske berör matematiska begrepp som du ännu inte stött på i mattekursen, men det borde gå bra att lösa dem ändå. I en annan laboration kommer vi att bekanta oss med ett annat matematikverktyg MATLAB. Du kommer under andra kurser att möta Maple och MATLAB och lär dig då successivt om systemen efter behov. Man lär sig naturligtvis mycket på att själv experimentera med systemen och prova saker som verkar kul.

Laborationstiden räcker säkert inte till för att göra allt, men det är heller inte meningen. Uppgifterna bör göras i angiven ordning och *-märkta uppgifter i första hand. Där ej annorlunda sägs finns erforderliga funktioner beskrivna i Gula Boken. Så se till att du har den till hands. Labhandledarna svarar naturligtvis också på frågor.

2 Maple i korthet

Maple är ett generellt matematikverktyg som är användbart i alla sammanhang där matematik spelar en roll.

Observera att tilldelningstecknet i Maple är "==" och att varje kommando skall avslutas med ";" (eller ":" för undertryckande av utskrift). Vid redigering flyttar man sig upp i den tidigare textmassan med musen (markören fixeras med **vänstra mustang-enten** inte med någon av de andra två, som har med klistring att göra) eller piltangenten och vid utförandet av ett tidigare kommando (eventuellt redigerat) ersätts det tidigare svaret. Med musen eller piltangent kan man återvända till slutet. Undvik att trycka på returtangenten om något i textmassan är markerat; det markerade tas nämligen som ett kommando. Observera också att liksom i UNIX betyder i Maple stor och liten bokstav olika sak (oftast).

3 Vi bekantar oss med Maple

*Uppgift 1. Start av Maple

Det kan vara praktiskt att först skapa en katalog (mapp) *maple*, vilket sker med UNIX-kommandot *mkdir maple*, och att med *cd maple* flytta sig till den katalogen. Starta sedan Maple med UNIX-kommandot *xmaple* eller med programmenyns *Maple* (i det sista fallet anser Maple att du är kvar i hemkatalogen). Om Maple-fönstret och arbetsbladet blir små är det idé att omedelbart förstora dem (ändring sparas i filen *.xmaplev4rc*).

*Uppgift 2. Uttryck

Skriv in följande rader (bortsett från kommentarerna och redotecknen >). Varje kommando kommer att resultera i ett svar som ej medtagits här.

```
> s:=2+x;           # Definiera ett uttryck
> x:=1;             # Definiera x
> s;                # Vilket värde har s nu?
> t:=2+x;           # Definiera ett uttryck till
> x:='x';           # Odefiniera x
> s;                # Vilket värde har s nu?
> t;                # Och t?
```

Notera att *s* hela tiden står för uttrycket $2+x$, som dock evalueras (beräknas) fullt ut om *x* är definierat, medan *t* - eftersom *x* var definierat när *t* definierades - står för uttrycket 3.

*Uppgift 3. Uttryck

Beräkna π med 100 siffror. Beräkna det därefter även med 200 siffror genom att med musen flytta dig till uttrycket, bara ändra ett tecken i det inmatade och till sist trycka på returangenten.

*Uppgift 4. Uttryck

Skriv in följande rader (bortsett från kommentarerna och redotecknen >). Varje kommando kommer att resultera i ett svar som ej medtagits här.

```
> 2+1/3; sqrt(99);   # Beräkna värdet av två uttryck
> 2+0.33;sqrt(99.0); # Två värden till
> 2+Pi; sqrt(99.0+Pi); # Och ytterligare två
```

Hade vi matat in motsvarande uttryck på en normal miniräknare, hade vi i samtliga fall fått numeriska approximationer. Men här får vi det bara i vissa fall. Vad kan det bero på? Beräkna - utan att skriva om något - numeriska värden i de saknade fallen.

*Uppgift 5. Lär dig redigera och bli imponerad

- Kontrollera att *x* är odefinierad genom att skriva ut värdet. Odefiniera, dvs skriv *x:='x'*; om så erfordras.

- b. Definiera uttrycket
 $s := x^8 + 12x^7 + 71x^6 + 260x^5 + 528x^4 + 388x^3 - 261x^2 - 540x - 459$;
Om det råkar bli fel (lätt hänt när man skriver av sådana här långa konstiga saker), använd då piltangenterna eller musen för att korrigera (se inledningen ovan). I musfallet flyttar du markören till rätt plats, klickar med vänstra musknappen och rättar.
- c. Skriv ut uttryckets värde.
- d. Faktorisera uttrycket med *factor*. Resultatet skall om du gjort rätt bli snyggt. Vad blev det? _____
- e. Ändra i uttrycket sista termen till 460 (använd musen för redigeringen) och tryck på returtangenten. Den nya definitionen kopplas till *s*.
- f. Flytta musen till *factor*-raden och tryck på returtangenten. Nu försöker Maple faktorisera det nya uttrycket, men "misslyckas" eftersom uppgiften är "omöjlig". Funktionen *factor* försöker hitta faktorer av lägre gradtal med rationella koefficienter. Speciellt hittas linjära faktorer bara om det finns rationella rötter, d v s av formen p/q , där p och q är heltal.
- g. Med ovanstående teknik försvinner de tidigare resultaten. Säg att vi vill se både den misslyckade faktoriseringen för 460, som ju syns nu, och den lyckade för 459 utan att behöva skriva om. Då kan vi ta till kopiering/klistring. Markera uttrycksberäkningen, klicka med vänstra mustangenten vid sista redotecknet, klicka med mellersta mustangenten, ändra 460 till 459 och tryck på returtangenten och vips har vi ett nytt uttryck. Alternativt kan du använda **Edit**-menyns **Copy/Paste**. Gör sedan på samma sätt med faktoriseringen.
- h. Med tekniken i g. arbetar vi i två steg. Man kan göra uttrycksberäkningen och faktoriseringen i ett svep genom att skriva de två kommandona efter varandra på samma rad eller på två rader uppdelade med SHIFT-RETURN. Faktorisera på det viset uttrycket med 460 ersatt av 219 genom kopiering/klistring. Hur många faktorer blev det? _____

Ett alternativ till klipp-och-klistra-tekniken är att stoppa in en variabel i den delen av uttrycket som vi vill kunna ändra. Se *tex* Uppgift 10.

*Uppgift 6. En formel

- a. Tag fram en så enkel formel som möjligt för summan $\sum_{k=1}^N k^5$. Använd *factor*. Vad blev formeln? _____
- b. Hur många siffror finns det i summans värde för $N=7000$? _____
Tips: använd funktionen *length* (skriv `?length` så får du reda på vad den gör; tag bort hjälpfönstret med vänstra rubrikknappens **Close**).

*Uppgift 7. Grafisk bestämning av nollställe

Ett enkelt och naturligt sätt att bestämma en reell rot till en ekvation $f(x)=0$ är att rita upp uttrycket $f(x)$ på ett allt snävare intervall kring den intressanta roten, vilket ju lätt låter sig göras i Maple (använd `plot(f(x), x=a..b)`; se som vanligt till att x är odefinierad). Bestäm med denna teknik roten i närheten av -2 till $x^3+2x^2+x+1=0$ på 0.001 när. Svar: _____

*Uppgift 8. Uttryck och funktioner

Ett uttryck definieras enligt modellen $u:=2+3*x$. En motsvarande funktion i stället som $U:=x->2+3*x$ (OBS! Skriv inte $U(x):=$, som kan ställa till många underligheter). Vill vi beräkna uttryckets värde för $x=7$ skriver vi $x:=7; u;$. Det är enklare att beräkna funktionens värde för $x=7$. Då skriver vi bara $U(7);$. $U(x)$ är naturligtvis ett uttryck. Verket *diff* arbetar på uttryck medan *D* bildar derivatan av en funktion (se sid 202 i GB). För att få utskrift av funktionsdefinitionen måste man skriva `print(U);`.

Skriv en Maple-funktion motsvarande $f(x)=x^4-1$. Kalla den `FUNK1`. Bilda även funktionens derivata och kalla den `DFUNK1`. Skriv till sist ut $f(3)$ och $f'(3)$.

*Uppgift 9. Numerisk bestämning av rot till ekvation

Givet en ekvation $f(x)=0$ med en reell rot finns det en käck metod att ta reda på den. Välj först ett tal x_0 i närheten av roten. Upprepa sedan följande beräkning till dess någon form av stabilitet synes uppstå:

$$\text{Ersätt } x_0 \text{ med } x_0 - f(x_0)/f'(x_0), \text{ där } f'(x) \text{ är derivatan av } f(x).$$

Metoden kallas Newtons iterationsmetod (iterera betyder upprepa) och är lätt att motivera, vilket vi dock inte går in på här.

I Maple gör du lämpligen ungefär så här. Definiera funktioner motsvarande f och f' . Sätt x_0 till lämpligt startvärde. Beräkna nytt x_0 -värde enligt formeln. Upprepa det sista steget ett antal gånger genom att bara klicka med musen och trycka på returtangenten.

- Genomför detta för ekvationen $x^3+2x^2+x+1=0$ och för den rot som ligger i närheten av -2 . Vilket värde stabiliseras beräkningarna vid? _____
- Maple har en funktion som heter *fsolve* för numerisk bestämning av rötter. Prova att hitta roten till ekvationen ovan med hjälp av *fsolve*.

*Uppgift 10. Uttryck och olika sorters problem

- Kontrollera att x och a är odefinierade. Definiera uttrycket $s:=x^3+a*x^2+x+1$;
- Skriv ut uttryckets värde.
- Sätt a till 1. Skriv ut uttryckets värde. Faktorisera det med *factor*.
- Sätt a till 2. Skriv ut uttryckets värde. Försök att faktorisera det med *factor*.
- Sätt a till $-37/3$. Skriv ut uttryckets värde. Faktorisera det med *factor*.

f. Sätt a till 2.1. Skriv ut uttryckets värde. Försök att faktorisera det med *factor*.

Vad lär vi oss av detta? I läroböcker brukar alla problem ha enkla lösningar. Men så är det uppenbarligen inte i verkligheten, inte ens när det gäller faktorisering. Faktoriseringfunktionen *factor* har de begränsningar som nämndes i Uppgift 5 och som torde sammanfalla med vår uppfattning om faktorisering. Visst skulle man kunna tänka sig att t ex ytterligare några typer av linjära faktorer skulle kunna plockas fram (exempelvis $x - \sqrt{n}$). Sammanfattningsvis: det finns problem som är i sig "omöjliga" och det finns andra problem som det aktuella systemet inte kan klara av.

*Uppgift 11. Integrera

Här följer några uppgifter med bestämda integraler från första delen av ett matematikkompendium (hösten 95), vilka (liksom alla de övriga) lätt löses med Maple. Gör en eller annan och jämför med facit-svaren.

$$80a. \int_0^{\pi/2} \sin^9 x dx \quad 96d. \int_0^{\infty} x^2 e^{-x} dx \quad 300. \int_0^{\pi} \frac{\sin x}{\cos^2 x + 4 \cdot \cos x + 7} dx$$

Uppgift 12. Integrera

I den övningsexempelsamling i matematik som användes när jag började på Chalmers hittar man bl a integralen $\int x^n e^{-x} dx$. Beräkna den för $n=3$ med Maple. Svaret kan förenklas med *factor*(""); En fråga att fundera över: Varför klarar inte Maple integralen för allmänt n ?

Uppgift 13. Integrera

Att beräkna obestämda integraler till rationella funktioner är i de enklaste fallen bara fråga om rutinarbete, eftersom det bygger på (se mattekompndiet) en partialbråksuppdelning av den rationella funktionen. Verkligheten är dock något brutalare, eftersom det inte alltid går att hitta en partialbråksuppdelning av den form man brukar se på i läroböckerna. Men Maple klarar allt (exakt om rationella koefficienter)!

Beräkna med Maple $\int \frac{x^2}{x^3 + ax^2 + x + 1} dx$ för några olika a -värden (jfr Uppgift 10). Lämpligen införs först en variabel för uttrycket (se till att x och a är odefinierade).

- $a=1$.
- $a=2$. I detta fall får man en komplicerad exakt formel som är kopplad till rötterna till en ekvation. Beteckna denna med *integral*. Då ger *evalf(integral)* en numerisk approximation.
- $a=2.1$. I detta fall får man direkt en numerisk approximation.
- $a=21/10$. Exakt formel.
- Beräkna ett par av motsvarande bestämda integraler för gränserna 0 och 1.

4 Grafik

*Uppgift 14. Bakgrundsinformation om grafik (mer än vad som finns i Gula Boken)

Kurvor och ytor kan matematiskt vara givna på tre olika former:

1. **Explicit form**, dvs (kurva) $y=f(x)$ med $x \in [a,b]$ respektive (yta) $z=f(x,y)$ med $x \in [a,b], y \in [c,d]$.
2. **Implicit form**, dvs $f(x,y)=0$ respektive $f(x,y,z)=0$.
3. **Parameterform**, dvs (kurva) $x=x(t), y=y(t)$ med $t \in [a,b]$ respektive (yta) $x=x(t,u), y=y(t,u), z=z(t,u)$ med $t \in [a,b], u \in [c,d]$.

Man använder i Maple för respektive fall (*implicitplot*, *implicitplot3d* och *spacecurve* ligger inte i standardbiblioteket utan kräver att man först skrivit `with(plots);`)

- 1 (kurva): `plot(f(x), x=a..b);`
- 1 (yta): `plot3d(f(x,y), x=a..b, y=c..d);`
- 2 (kurva i 2D): `implicitplot(f(x,y)=0, x=a..b, y=c..d);`
- 2 (yta): `implicitplot3d(f(x,y,z)=0, x=a..b, y=c..d, z=e..f);`
- 3 (kurva i 2D): `plot([x(t), y(t)], t=a..b);`
- 3 (kurva i 3D): **OBS! Annorlunda än för 2D**
`spacecurve([x(t), y(t), z(t)], t=a..b);`
- 3 (yta): **OBS! Annorlunda än för 2D**
`plot3d([x(t,u), y(t,u), z(t,u)], t=a..b, u=c..d);`

Vill man rita flera kurvor/ytur låter man funktionerna/uttrycken bilda en mängd, t ex `plot({f(x), g(x)}, x=a..b);`

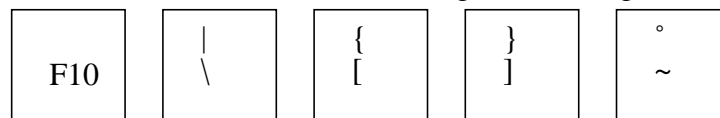
eller

`implicitplot3d({f(x,y,z)=0, g(x,y,z)=0}, x=a..b, y=c..d, z=e..f);`
eller när det gäller parameterformen `plot([...], [...])` respektive `plot3d([...], [...], t=a..b, u=c..d);`

Genomgående är implicit-rutinerna långsammare än övriga. I dessa kan man för övrigt ha ett godtyckligt uttryck i x, y och z i stället för $f(x,y,z)=0$.

*Uppgift 15. Var finns tecknen [,], { och }?

Dessa tecken behövs snart. Numera har tangentborden på E alla intressanta tecken (några finns på märkta eller omärkta funktionstangenter till höger om F10 på tangent-



bordet) och när man trycker på motsvarande tangent blir det rätt på skärmen. Precis som det skall vara! Detta under förutsättning att standardinställningarna inte ändrats (det som skall gälla är mellersta bakgrundsmenyens **Keyboard Options/Key Swe Ext**).

*Uppgift 16. Rita i 2D

- Rita kurvan $y=\sin(x)/x$ på intervallet $[-10,10]$. Prova de olika alternativen i menyn **Axes** (tryck först på figuren så att den är vald) och se efter vad de har för effekt. Förminska/förstora den genom att dra i någon av de små fyrkanterna i markeringsramen.
- Tag med hjälpmenyns **Topic Search** (sökord `color` och sedan **Apply** eller **OK**) reda på hur du kan få kurvan i a) att bli gul. Du kan alternativt använda `?plot` följt av en tryckning på `color` längst ned på den hjälpsida som då visar sig. Rita sedan upp kurvan gulfärgad. Det finns en mängd `s k` optioner till ritkommandona, vissa kan man ändra med menyer.
- Rita upp lissajouskurvan $x=\sin(4*t)$, $y=\sin(7*t)$ på intervallet $[0,2*\pi]$ (π heter så i Maple). Ändra eventuellt sedan 4 och 7 mot något annat.
- Rita upp ellipsen $x^2+4y^2=1$. Ellipsen ser ut som en ellips om du i **Projection**-menyn (tryck först på figuren) väljer **Constrained** istället för förvalda **Unconstrained**. **Tips:** `with(plots)` och sedan `implicitplot(...)`.

Uppgift 17. Rita flera funktioner i 2D

Låt $f(x)=\sin(x)/x$. Rita upp kurvorna $y=f(x)$, $y=f'(x)$ och $y = \int_0^x f(t)dt$ på intervallet $[-20,20]$ i en figur.

Uppgift 18. Visualisera taylorutveckling

Bestäm polynom av grad 1,3,5,7 och 15 som approximerar funktionen $\sin(x)$ i närheten av $x=0$ (se *taylor* och *convert*, sid 201). Lägg för enkelhets skull polynomen i fem olika variabler, t ex $p1$, $p3$, $p5$, $p7$ och $p15$. Rita sedan upp dessa funktioner och $\sin(x)$ i ett diagram (se sid 176, mitten, eller uppgift 15) för intervallet $[0,10]$. Det är lämpligt att i `plot`-kommandot lägga till en sista parameter som anger önskat y -intervall, t ex `-10..10`.

*Uppgift 19. 3D-ritning

- Hitta på en trevlig funktion $f(x,y)$ av två variabler (eller tag i brist på fantasi $f(x,y)=\sin(\sqrt{x^2+y^2})/\sqrt{x^2+y^2}$). Rita upp den och prova menyerna **Style**, **Color** och **Axes** (tryck först på figuren). Tryck på högra musknappen eller R-knappen för att få omritning. Vrid på figuren med vänstra musknappen. **Ledning:** Se Uppgift 14.
- Rita en "igloo" med följande kommando **utan att mata in det** för hand (ytan är på parameterform, dvs en funktion för x , en annan för y och en tredje för z) `plot3d([t*sin(t)*cos(u),t*cos(t)*cos(u),t*sin(u)],t=0..2*Pi,u=0..Pi)`; Ge i stället kommandot `?plot3d` och kopiera och klistra in kommandot från exempeldelen av hjälpen (man kallar t för x och u för y men det spelar ju ingen roll)! Vrid på igloon med vänstra musknappen. Se till att ytorna får färg.

- c. Rita upp sfären $x^2+y^2+z^2=4$. Rita sedan upp denna sfär och planet $x=1$. Vrid och vänd.

5 Dokumentation

*Uppgift 20. Dokumentation

Man kan foga in figurer från Maple i FrameMaker-dokument. Hur det går till beskrivs i Gula Boken. Men man kan - som omtalas i kompendiet - också göra sin dokumentation direkt i Maple eftersom det går att stoppa in bl a förklarande text (även svenska bokstäver tycks fungera numera) i ett arbetsblad. Det finns åtskilliga möjligheter. Här ser vi bara på en.

Markera en sektionssklammer i arbetsbladets vänsterkant. Välj **Insert/Text Input**.

Skriv in text i den nybildade sektionen. Kontextraden (se figuren på sid 192 i GB) byter samtidigt utseende (sid 211). Prova eventuellt att välja annan stil än den normala med den vänstra menyn på kontextraden eller använd någon av de följande. Ett exempel ser du till höger.



The next example illustrates

- plotting
- colors

```
> plot(sin(x), x=0..30, color=green);
```

I **Edit**-menyn finns en mängd alternativ för redigering av arbetsbladet, t ex för att ta bort delar, så man kan lätt avlägsna sina misstag före utskrift utan att börja på nytt.

6 Några problem från "verkligheten"

Uppgift 21. Tillverka sekvenser

Man vill ofta tillverka sekvenser med någon form av regelbundenhet, t ex $1, 2, 3, \dots, N$ eller $f(1), f(2), \dots, f(N)$, där N är givet och f en given funktion. Detta kan göras med `seq`(uttryck som innehåller $i, i=1..N$). Som vanligt måste i (eller det namn vi använder) vara odefinierad eller `'...'`-notation användas (jfr *sum* och *plot*, sid 210). T ex

```
> seq(i, i=1..5);  
1, 2, 3, 4, 5  
> seq(1, i=1..5);  
1, 1, 1, 1, 1
```

Tillverka de fyra följande sekvenserna (gör det utan att skriva om något i onödan!):

- $1, 8, \dots, N^3$ med $N=10$
- $\sin(0), \sin(0.01), \dots, \sin(6.28)$
- $[1, 1], [2, 4], \dots, [N, N^3]$ med $N=10$
- $[0, \sin(0)], [0.01, \sin(0.01)], \dots, [6.28, \sin(6.28)]$

och rita upp punkterna i d. i ett punktdiagram (se GB, sid 206).

Uppgift 22. Lotto

Vi vill generera 7 slumpmässiga heltal mellan 1 och 35. Tag först med hjälpmenyns **Topic Search** reda på vad xxxx nedan skall ersättas med för att följande två rader skall generera två slumpetal i det intervallet (tips: sökord random+**Apply**).

```
> slump:=xxxx(1..35);      # Skapar en funktion för slumpetal
> slump();slump();        # som vi sedan kan anropa lika många
                           # gånger som det önskade antalet
                           # slumpetal
```

Skapa sedan en följd av 7 slumpmässiga heltal (ej nödvändigtvis olika) mellan 1 och 35.

Uppgift 23. Kobingo med 200 kor och vinster

Vi vill generera ett antal slumpmässiga punkter med heltalskoordinater mellan 1 och 50. T ex om antalet är 2 (5,34),(23,19). Producera med Maple en sekvens av 200 sådana punkter och rita dessutom upp dem i ett punktdiagram (se GB, sid 206). Vad händer om man glömmer style=POINT i plot-kommandot?

Uppgift 24. Rotritning

Att rita upp nollställena till ett polynom är lätt i MATLAB, som du kommer att märka senare. I Maple är det omständligare (jag utesluter dock inte att jag missat någon procedur som gör det enkelt). Skriv Maple-kommandon som ritat upp rötterna till $x^{34}=1$. **Ledning:** Vi vet från Gula Boken (sid 205) att *fsolve* ger en sekvens z_1, \dots, z_{34} av rötterna och att `plot([[x1,y1], ..., [xn,yn]], style=POINT)` ritat punkter. Problemet är då att göra om sekvensen till en lista med rötterna på formen $[x,y]$. Vi kan bilda en sekvens R som är just $[x_1, y_1], \dots, [x_n, y_n]$ med `R:=seq([Re(s[i]), Im(s[i])], i=1..gradtalet)`. Sedan plottar vi bara listan $[R]$.

7 Maple på webben

Följande webbadresser nämns i Gula Boken:

- <http://www.indiana.edu/~statmath/math/maple/> — diverse Maple-info från Indiana University.
- <http://www.maplesoft.com/> — information om Maple och annat från företaget som säljer Maple.

Laborationer i kursmomentet Datoranvändning E1

Laboration nr 4: Dokumentproduktion med FrameMaker

NB: Filer som omtalas finns tillgängliga vid första laborationstillfället

96-09-18: Magnus Bondesson

Uppdatering 98-10-28: Thomas Hallgren

1 Introduktion

Denna laboration handlar om dokumentproduktionssystemet FrameMaker, som finns på E-sektionens arbetsstationer. FrameMaker (försvenskat: Ramskaparen) är principiellt sett mycket likt t ex Word för PC och Mac. Idag tar vi upp vad som behövs för att åstadkomma enklare dokument med ett standardiserat utseende. Vid ett senare tillfälle går vi in på några mer avancerade ting.

Som tidigare markerar symbolen i uppgifter med flera delar inledningen på ett stycke med en konkret arbetsuppgift. Dessförinnan kan finnas motiverande eller förklarande text.

Innan du kommer till laboration 4 skall du ha läst eller skummat kapitlet om FrameMaker i Gula Boken. Hoppa som vanligt bara över det bland dessa sidor du tycker är konstigt eller för tillfället ointressant och återvänd i stället efter laborationen.

Om du inte hann med alla uppgifterna på Laboration nr 3

Det gjorde nog ingen. Vissa av restuppgifterna kommer vi att utnyttja vid ett senare laborationstillfälle.

2 Uppgifter

Uppgift 1. Starta FrameMaker

Starta FrameMaker via bakgrundsmenyn med program. Det finns två alternativ: engelsk eller svensk version. Välj den svenska versionen så att bilderna i Gula Boken stämmer överens med vad som visas på skärmen (bokens bilder är producerade på SUN-datorer med ett något annorlunda fönstersystem, vilket förklarar varför rubrikerna ser litet olika ut på din skärm).

Efter en stund dyker ett litet fönster upp på skärmen med fem knappar. Låt oss titta på dem från höger:



- AVSLUTA Avslutar FrameMaker, så den knappen provar vi inte nu.
- INFO Ger information om version av FrameMaker etc. Prova.
- HJÄLP Öppnar ett fönster där du kan söka information om FrameMaker. Prova att klicka på text, pilar och figurer och se vad som händer. Vi kommer nog oftast att ta oss in i hjälpsystemet på annat sätt.
- ÖPPNA Använder du då du skall öppna ett befintligt dokument, som du vill redigera eller utgå ifrån. När du klickar på knappen får du upp ett fönster som visar katalogerna (överst) och filerna i din hemkatalog.
- NYTT Använder du då du skall skapa ett helt nytt dokument och inte har något tidigare att utgå ifrån. När du klickar på knappen får du upp ett fönster där du kan välja bland färdiga mallar.

Uppgift 2. Titta på ett färdigt dokument

- Öppna med **ÖPPNA**-knappen det dokument som du nu håller i din hand. Det finns som `~hallgren/Intro/lab_4.doc`. Vid det här laget har du mött öppningsdialoger i några program. FrameMaker har sin egen variant. Du kan antingen skriva in det fullständiga filnamnet (slutdelen räcker om du är i rätt katalog) överst i dialogen eller klicka dig fram. FrameMaker har inte en ruta för kataloger och en annan för filer, utan i stället räknas först alla kataloger upp och följs av filerna.
- Prova nu att
- Flytta dig mellan sidorna med hjälp av hissen och knapparna i fönstrets nederkant.
 - Förstoring på olika sätt.
 - Prova med olika storlekar på fönstret att välja **Anpassa sidan till fönstret**.
 - Prova att ställa in lagom förstoring (t ex 100%) och välj sedan **Anpassa fönstret till sidan**.
 - På sidorna ser du prickade inramningar och diverse styrtecken (som inte har med den egentliga texten att göra och inte skrivs ut), speciellt styckeslut ¶, som markerar avslutning på ett stycke. Varje stycke har sina egna egenskaper (styckeutformning) och dessa beskrivs med ett styckenamn (styckeutformningsnamn vore korrektare, men tyngre). Flytta markören till stycket efter rubriken för uppgift 3. Styckenamnet kan avläsas i överdelen och underdelen av fönstret. Vad heter stycket? _____
Flytta sedan markören till stycket som utgör rubriken för uppgift 3. Styckenamn? _____
Dito till laborationstiteln. Styckenamn? _____



- f. Styrtecknen (och även ramar m m) kan om man så önskar undertryckas. Använd menyn **Visa/Text symboler** för detta. Självt tycker jag att det är bra att alltid se dem, men smaken är ju delad.



Vi skall nu manipulera innehållet i dokumentet en aning.

- g. Flytta markören till en plats i den löpande texten. Skriv in litet text. Pröva **Redigera/Ångra**.
- h. Markera text med dragning. Kopiera med **Redigera**-menyns **Kopiera**, markera ny plats och klistra in med **Redigera**-menyns **Klistra in**. Pröva återigen **Ångra**.
- i. Dubbelklicka i ett ord. Hela ordet skall markeras.
- j. Trippelklicka i ett ord (dubbelklicka och strax därefter ett klick till). Hela stycket skall markeras.
- k. Placera markören i ett stycke med namnet *Brödtext* (löpande text brukar i den grafiska branschen kallas så). Tag fram styckekatalogen (knappen längst upp i figuren på sidan 4 i detta PM). Välj *Rubrik 1*. Stycket byter till den valda utformningen. Återgå till det ursprungliga med *Brödtext* eller **Ångra**.
- l. Markera en del av texten i vanlig stil någonstans. Tag fram teckenutformningskatalogen (andra knappen i figuren på sidan 4). Välj *Tecken fet*. Utformningen ändras. Återgå med *Standardtypsnitt* eller **Ångra**.



Du har nu gjort ett antal förändringar i dokumentet och skulle kunna spara det i den egna katalogen, men det är knappast värt besväret den här gången. Så stäng det bara med **Arkiv/Stäng** utan att spara.

Uppgift 3. Få mera begrepp om vad FrameMaker kan.



Tryck på **HJÄLP**-knappen. Välj *FrameMaker översikt* och bläddra dig fram till *Exempeldokument*. Titta på dessa: *Rapport*, *Meritförteckning* och *Figurer* genom att dubbelklicka på namnet. Stäng sedan fönstret med **Arkiv/Stäng**. Dessa dokument kan användas som utgångspunkt för egna. Figurer i *Figurer* kan markeras, kopieras och klistras in i egna dokument. Vi kan nå dessa exempel enklare via **Hjälpmenyn** i ett dokumentfönster.

Uppgift 4. Nu gör vi ett eget dokument

Du skall tillverka ett eget dokument utgående från en mall. Det finns många olika mallar, men avsikten är att du skall göra en rapport om litet om det du har upplevt hittills under datoranvändningskursen, så vi väljer en rapportmall. Rapporten skall inte lämnas in eller detaljgranskas, men det begärs att du skriver riktiga meningar och ord. Däremot får det gärna vara si och så med logiken och sammanhanget. **Dokumentet skall ha den struktur som visas på sista sidan.**

Den här gången lämnar jag inte alltid fullständiga arbetsanvisningar (utan hoppas att du minns litet från föreläsningen eller Gula Boken), så du får slå upp eller fråga en handledare. Tillbringa inte för mycket tid med att leta efter behövlig information på egen hand.

Innan vi sätter igång ett par erinringar: Returtangenter ger övergång till nytt stycke. ENTER-tangenten (längst ned till höger) ger manuell radbrytning inom ett och samma stycke, men det normala är ju att låta FrameMaker sköta radbrytning automatiskt. Fönster av olika slag som dyker upp och inte har någon egen **Avsluta-** eller **Stäng-**knapp kan stängas genom att man dubbelklickar på knappen längst upp till vänster. Och kanske viktigast: om du märker att du gjort ett misstag, tag det lugnt. Med **Redigera**-menyns **Ångra** kan närmast föregående åtgärd av nästan vilket slag som helst ångras. Men bara närmast föregående, så gör inget förhastat. Det lätt återgå till den senast sparade versionen av dokumentet (med kommandot **Arkiv/Återgå till sparad**) så det kan vara en bra ide att spara dokumentet då och då, när det är i ett bra skick.

- Välj **NYTT**-knappen. Titta eventuellt mycket hastigt på några exempel med färdiga mallar med hjälp av **Standardmallar**-knappen i den dialog som visar sig. T ex på Brev, Rapport normal och Nyhetsblad. Avsluta varje exempel med **Arkiv/Stäng**.
- Välj **NYTT**-knappen igen. Öppna mallen *Rapport normal*.
- Skriv in rapportens rubrik (styckenamnet *RapportTitel*; avsluta med returtangenter) och gruppledammarnas namn (styckenamnet *RapportFörfattare*; avsluta med returtangenter). Som sagt använd vettig text.
- Du har nu fått början av ett stycke, som skall utgöra rubrik (styckenamnet *Rubrik 1*). Fyll i rubriken och tryck på returtangenter.
- Nu har vi hamnat i början av ett stycke med löpande text (styckenamnet *Text*). Skriv in ett par sådana stycken. Avsluta det sista med returtangenter.
-  Om vi fortsätter att skriva fortsätter vi med löpande text, men se det vill vi inte, utan vi vill ha en rubrik av samma slag som den tidigare (styckenamnet *Rubrik 1*). Tag därför fram styckekatalogen med översta knappen i figurerna på ömse sidor (svensk till vänster, engelsk till höger). Och välj i katalogfönstret *Rubrik 1*. Notera att styckekatalogen är annorlunda än förut. Tyvärr är mallarnas stycke- och teckenkataloger rätt fattiga. 
- Skriv in ny rubrik och sedan ytterligare ett par stycken med löpande text. Kom ihåg att spara dokumentet då och då (t ex nu).
- Producera ytterligare två sådana här numrerade grupper. Du kan göra det på samma sätt som i förra momentet, men alternativt genom att du markerar med musen, kopierar med **Redigera/Kopiera**, markerar insättningsplats med musen och klistrar in med **Redigera/Klistra in**.
- Prova **Kontrollera stavning** under **Redigera**-menyn. (Detta kommando skulle säkert kunna ägnas bortåt en timme, men gör inte det.)
- Prova **Synonymordbok** under **Redigera**-menyn.
- Rita nu en bild med FrameMakers ritverktyg (3:e knappen i figurerna ovan) som visar hur det ser ut när du sitter vid en datorskärm. Se avsnitt 7.5 (Rita figurer) i Gula Boken. Observera att du först måste markera en insättningsplats och skapa en kopplad

ram (avsnitt 7.5.1-7.5.3). Notera att verktygsmenyn kan expanderas (avsnitt 7.5.4). Eventuellt kan du hämta någon del av bilden från den i Uppgift 3 nämnda figursamlingen.

- Hämta nu in en färdig bild (lagrad som ~hallgren/Intro/OS.gif, i GIF-format), som visar något av enkätresultaten. Placera först markören någonstans i texten. Använd sedan **Arkiv**-menyns **Importera/Fil**. Välj Kopiera till dokumentet och sedan 300 dpi (motsvarande bildstorlek anges till höger). Du kan alltid i efterhand skala om bilden med t ex **Bild**-menyns **Skala**. Bilden ser tyvärr nog inte så bra ut på skärmen men blir OK vid utskrift (om än något liten).
- Vi vill att text skall kunna löpa på ena sidan om bilden ovan. Markera bildramen och välj **Special**-menyns **Kopplad ram**. Ändra i dialogen (se figur sid 137) **Placering** till **Att gå in i stycket** och **Justering** till **Vänster** eller **Höger** och tryck på **Ändra ram**.
- Gör sedan en tabell som ser ut ungefär som följer (påhittade värden). Tabeller står det inte något om i Gula Boken, men vi får se om det inte går lätt ändå med **Tabell**-menyns **Sätt in tabell**.

OS	Antal
DOS	10
Windows	30
95	
UNIX	1
MacOS	7

- Nu vill vi ha in en bild av hur det ser ut på skärmen, t ex en bild av FrameMakers startfönster med de fem knapparna nu inklusive fönsterrubrik. Använd **Arkiv/Hjälpmedel/Avbilda+Arkiv/Importera/Fil** för detta.
- Allra sist skall det naturligtvis i rapporten omtalas att du sysslat med Maple. Och tala om att Maple kan beräkna såväl

$$\sum_{k=1}^N k^5 \text{ som } \int_0^{\pi} \frac{\sin x}{\cos^2 x + 4 \cdot \cos x + 7} dx$$

Läs i avsnitt 7.6 om hur du går tillväga för att skapa formler.

Uppgift 5. Utfyllnad vid behov

Förr eller senare behöver du skriva π i texten eller ordet idé, så det är lika bra att veta hur det går till eller rättare sagt hur man tar reda på hur det går till. Använd **Hjälpmenyns Teckenuppsättningar** för detta. Se även avsnitt 7.9.1 i boken.

Uppgift 6. Skriv ut dokumentet

Ditt dokument borde nu vara tvärsidigt och det tycker jag att vi kan unna oss att skriva ut, så gör det. Men låt först handledaren titta på dokumentet vid skärmen. Välj **Arkiv/Skriv ut**. I utskriftsdialogen brukar jag alltid ange vilka sidor jag vill ha utskrivna (snarare än Alla). Det kan också vara bra att se till **Sista sidan först** inte är valt. Hämta utskriften (i rutan "Skrivare:" står det antagligen \$PRINTER. Vilken skri-

vare utskriften då hamnar på får du reda på med UNIX-kommandot `echo $PRINTER`). Du kan läsa mer om utskrifter på avsnitt 7.3.8.

Uppgift 7. Följ en inbyggd självstudiekurs

FrameMaker har ett inbyggt hjälpsystem, vilket vi redan mött. Det finns också en självstudiekurs (eng. tutorial) uppdelad i ett antal lektioner, som är tänkta att lära användaren olika delar av programmet. Det fina med denna är att man får fin interaktivitet: eleven följer förklaringar och anvisningar och får omedelbar respons på sina åtgärder (både om de är korrekta eller fel). Däremot är det en smaksak om man gillar att läsa sådant här på skärmen eller på papper. Självstudiekursen är skriven helt i FrameMaker och utnyttjar de hypertextmöjligheter och andra finesser som ingår, men vilka vi inte planerar ta upp i denna kurs.



Starta självstudiekursen med **Hjälp**-menyns **Självstudiekurs** (eller från Hjälp-knappen) och följ sedan Lektion 1. FrameMaker vill att alla öppna dokumentfönster stängs först, så låt så ske. Lektionen skall ta 30 minuter, men jag hoppas att du är raskare än så.

Uppgift 8. Avsluta

Avsluta FrameMaker med AVSLUTA.

RAPPORT OM DITT OCH DATT

Magnus Bondesson

1.0 Rapportens syfte

Att beskriva

2.0 Först surfade vi

Det var rätt kul. Men vi upptäckte också att det faktiskt kunde tjäna ett vettigt syfte.

3.0 Sedan blev det UNIX

Liknar ju delvis DOS!

4.0 Matematik lätt som en plätt

Maple lät oss räkna ut π med 200 decimaler utan vidare.

5.0 Och nu ordbehandlas det

Verkar svårt.

Laborationer i kursmomentet Datoranvändning E1

Laboration nr 5: Matteverktyget MATLAB

92-11-05, 93-11-01, 94-10-31, 95-10-09, 96-10-31, 97-11-13 Magnus Bondesson
Uppdatering 98-10-26 Thomas Hallgren

1 Introduktion

Syftet med denna laboration är att ge dig en första praktisk kontakt med matematikverktyget MATLAB. Precis som i Laboration nr 3 (som handlade om Maple) avser vissa av uppgifterna dessutom att antyda hur dessa verktyg kan användas för att ge ökad förståelse kring något fenomen.

Som förberedelse måste du ha läst igenom MATLAB-kapitlet i Gula Boken (hoppa som vanligt över det som verkar konstigt). Gör uppgifterna i ordning, i första hand de uppgifter som är *-märkta. Några av uppgifterna utan * kräver kanske lite grundläggande kunskaper i programmering.

Det är lätt att blanda ihop Maples beteckningar med MATLABs. Observera speciellt att tilldelningstecknet i MATLAB är "=" och att kommandon normalt inte avslutas med ";" (";" undertrycker utskrift av resultatet). π heter `pi` och inte som i Maple `Pi`, etc. MATLAB arbetar till skillnad mot Maple inte i ett eget fönster, utan i ett vanligt terminalfönster. Man kan därför inte använda musen vid redigering, däremot fungerar piltangenterna. Figurer i egna fönster (man kan ha flera, men normalt återanvändes ett enda automatiskt).

2 Uppgifter

*Uppgift 1. Starta MATLAB

Det är lämpligt — men inte nödvändigt — att först skapa en katalog (mapp) `matlab`, vilket du gör med UNIX-kommandot `mkdir matlab`. Starta sedan MATLAB med `tex` ett programmeny-kommando.

*Uppgift 2. En inblick i MATLABs värld

Innan vi börjar med uppgifter som illustrerar de mer grundläggande funktionerna i MATLAB kan det kanske vara kul att ta en titt på lite mer avancerade exempel för att få en känsla för vad man kan använda MATLAB till.

Till MATLAB hör en massa demonstrationsprogram, som är skrivna i MATLAB (källkoden är tillgänglig). Du kommer in i demonstrationsdelen genom att ge kommandot `demo`. Gör det och pröva `tex Visualization/2D Plots`, `Visualization/3D Plots`, `Gallery/Hoops`, `Games/Minesweeper` och `Visualization/Lorenz attractor animation`.

*Uppgift 3. Vektorer, ritning och redigering

Vi skall rita upp kurvan $\sin(x)$ på intervallet $[0,16]$.

- Bilda en vektor $x = [0, 1.0, 2.0, \dots, 16]$ med litet skrivarbete.
- Bilda en vektor y med motsvarande sinus-värden.
- Rita kurvan med $\text{plot}(x,y)$.
- Resultatet blev inte alldeles lyckat eftersom vår diskretisering var väl grov. Ändra med piltangenternas hjälp (se sid 169) så att $x = [0, 0.1, 0.2, \dots, 16]$ i stället samt — även detta med piltangenterna — bilda motsvarande vektor y och rita på nytt.

*Uppgift 4. Addition och slumpstal

Rita upp kurvan $\sin(x)$ störd med slumpmässigt brus i

- intervallet $(0.0,0.4)$
- intervallet $(-0.2,0.2)$

på samma intervall som i Uppgift 3, dvs lägg till varje sinus-värde ett slumpmässigt tal i det angivna intervallet. Vektorn x har du kvar sedan förra uppgiften, så skriv bara ett plot -kommando. **Ledning:** Slumptalsfunktionen rand beskrivs i Gula Boken, sid 170.

Uppgift 5. Vektorer och Lissajous-kurvor

Rita upp kurvan $x = \cos 4t, y = \sin(3t)$ på t -intervallet $[0,2\pi]$. Konstanten π finns fördefinierad.

*Uppgift 6. Funktioner som arbetar på vektorer

Betrakta uttrycket $\frac{4}{1+x^2}$.

- Beräkna värdet för $x=0.5$.
- Beräkna i ett svep uttryckets värde för $x = 0, 0.1, 0.2, \dots, 1.0$.
- Rita upp motsvarande kurva från $x = 0$ till $x = 1$.

*Uppgift 7. Rötter

Bestäm rötterna till 16:e-gradsekvationen $17x^{16} + 16x^{15} + \dots + 3x^2 + 2x + 1 = 0$. Rita upp dem i ett diagram (markera varje rot med + eller *). **Ledning:** Funktionen roots beskrivs i Gula Boken, sid 172. Om du händelsevis hann med Uppgift 24 i Laboration nr 3, notera hur mycket enklare detta problem löses i MATLAB än i Maple.

Uppgift 8. Rotvandring

Studera hur rötterna vandrar när nolltegrads-termen i förra uppgiften avtar från 1 till -1 , genom att i samma diagram rita in rötterna (med markering `.`) när nolltegradstermen är 1, 0.9, 0.8, ..., -0.9 , -1.0 . Gör först *hold on* för att få ritning i samma diagram och ändra sedan nolltegradstermen successivt i en FOR-snurra (eller arbeta manuellt). Vad händer med de två rötter som ursprungligen låg längst till vänster i diagrammet? Hur många reella rötter har utgångsekvationen respektive slutekvationen? Gör *clf* (av Clear Figure) eller *hold off* innan du fortsätter med en ny uppgift.

Uppgift 9. Bild från MATLAB fogas in i FrameMaker eller HTML-dokument

Spara diagrammet från någon av de två föregående uppgifterna som en PostScript-fil MATTE.epsi (se sid 180). Kontrollera resultatet med t ex

```
ghostview MATTE.epsi.
```

Anm. Från Laboration nr 4 vet vi hur bilder kan importeras till FrameMaker. Även PostScript-bilder kan importeras.

När det gäller HTML är det numera en aning värre. Förut kunde man från MATLAB spara på GIF-format, som ju hanteras direkt av webbläsare, men man tycks ha övergivit det formatet för gott. I Gula Boken (sid 181 överst) står hur man kan göra. Det sättet fungerade inte när texten skrevs. Nu går det, men dels blir PPM-filen väldigt stor, dels utvidgas bilden till A4-format. Bättre är att i MATLAB skriva

```
print -dppmraw filnamn
```

och att senare med `xpaint` eller `xv` ta bort onödig utfyllnad i `filnamn.ppm` och till sist spara på GIF-format. I version 5.1 (som ännu inte finns på E) kan man även spara på JPEG-format - som webbläsarna ju också klarar - med (*help print* ger detaljerna) `print -djpeg filnamn`. Bilden blir dock stor (som A4).

*Uppgift 10. Komplexa tal

Låt $z = 1 + 2i$. Beräkna $z^3 + z^5$. Beräkna sedan i ett svep detta uttryck för $z = 1 + 2i, 1 + 3i, \dots, 1 + 9i$.

Uppgift 11. Komplexa vektorer

Låt z vara en vektor av punkter som ligger på enhetscirkeln i det komplexa talplanet. En sådan vektor kan bildas t ex så här:

```
t=0:0.1:2*pi; z=cos(t)+i*sin(t).
```

Var hamnar då de punkter som utgörs av $w = \frac{1-z}{1+z}$? Vill du visualisera resultatet är det lämpligt att förfara enligt (>> är redotecknet som ej skall skrivas):

```
>> axis([-10,10,-10,10]); % Skala på axlarna
>> plot(z, '.')           % Rita punkterna på cirkeln
>> hold on                % Rita utan att sudda föregående
>> plot(uttryck för w, '.') % Rita de nya punkterna
```

```
>> clf % Innan du går vidare
```

Svar: _____

Uppgift 12. Summation

Beräkna följande summor numeriskt. (**Ledning:** *sum*, se sid 171.)

a. $\sum_{k=1}^{60} \frac{1}{k^4}$. Jämför gärna med summan $\frac{\pi^4}{90}$ för motsvarande oändliga serie.

b. $\sum_{k=1}^{30} \sin \frac{1}{k}$. Svar: _____

*Uppgift 13. Egna funktioner. Integration.

- a. Tag med hjälp av *help* eller *helpdesk* reda på vad verktygslådefunktionen *quad* gör (läs bara de 5 första raderna). Beräkna sedan en approximation till integra-

len $\int_0^{\pi/2} \cos x dx$ och jämför med det exakta värdet 1.

- b. Skriv en MATLAB-funktion *first*, dvs gör en m-fil (sid 183) *first.m* utanför MATLAB med ditt favoritredigeringsprogram (tex NEdit), som beräknar $\frac{4}{1+x^2}$ för en **godtycklig vektor** *x*. Spara lämpligen i katalogen `matlab` som du skapade inledningsvis. En sådan funktion har formen

```
function y=first(x)
y=?????
```

Testa den och verifiera att den ger rätt resultat.

- c. Beräkna nu med *quad* en approximation till integralen $\int_0^1 \frac{4}{1+x^2} dx$ och jämför med det exakta värdet som ju är π .

Uppgift 14. Egna funktioner. Ekvationslösning.

- a. Tag med hjälp av *help* reda på vad verktygslådefunktionen *fsolve* gör. Beräkna sedan en approximation till det nollställe till $\cos x$ som ligger i närheten av 1.5. Jämför med det exakta värdet som ju är $\pi/2$.
- b. Funktionen $5xe^{-x} - 1$ har två reella nollställen på intervallet $[0,5]$. Visa det genom att rita upp funktionen.

- c. Skriv en MATLAB-funktion *second*, dvs gör en m-fil *second.m* utanför MATLAB med ett redigeringsprogram, som beräknar funktionsvärdena för en godtycklig vektor. En sådan funktion har formen

```
function y=second(x)
y=?????
```

- d. Beräkna nu med *fsolve* det största av de två nollställena i b. Vad blev det?

Uppgift 15. Egna funktioner. Interaktion.

Skriv en MATLAB-funktion *inter*, dvs gör en m-fil *inter.m*, som låter dig markera 10 punkter i ritfönstret (som förutsätts finnas framme) och när en punkt markerats förbinder den med närmast föregående (utom för den första). En sådan funktion har formen

```
function inter
[xold,yold]=ginput(1);
for i=1:9
    ???
end
```

där ??? står för maximalt 4 rader kod.

Gör *clf* (suddar ritfönstret) och *hold on* och testa sedan din funktion.

*Uppgift 16. Ett besök till i fraktalernas underbara värld

Låt oss välja en punkt (x_0, y_0) i x, y -planet och sedan bilda nya punkter enligt formeln,

$$\begin{cases} x_{n+1} = 1 - y_n + |x_n| \\ y_{n+1} = x_n \end{cases}$$

för $n = 0, 1, 2, \dots$. Var hamnar dessa punkter? Följande funktion i form av en m-fil gör 1000 iterationer utifrån en startpunkt och ritar samtliga punkter. Skriv in funktionen (texten som inleds med % kan du strunta i) med valfritt redigeringsprogram och spara den som filen *ginger.m* i din katalog *matlab*.

```
% GINGER(x,y) draws PEPPARKAKSGUBBEN if e g x=-1.3,y=2
function ginger(x,y)
clf                                %erase previous figure and forget hold
axis([-10,10,-10,10])              %our coordinate system
plot([-10,10],[0,0])                %draw x-axis
hold on                             %never erase,but autoscale still in effect
plot([0,0],[-10,10])               %draw y-axis
for i=1:1000                         %iterate and plot 1000 times
    x1=1-y+abs(x);y=x;x=x1;
    plot(x,y, '.' )
end
```

Anropa den med *ginger(-1.3,2)*. Experimentera med andra startvärden. Resten är klar överkurs. Ändra i funktionen *abs(...)* till *sqrt(abs(...))* eller *sin(...)*. Gör eventuellt om

funktionen så att startpunkt väljs interaktivt med musen (jämför med föregående uppgift).

Uppgift 17. 3D-ritning

Hitta på en trevlig funktion $f(x,y)$ av två variabler. Skriv den som en m-fil och rita upp den med specialfunktionen *myplot3d* på ett uttryck (se sid 175 och sid 186) eller gör stegen i den funktionen.

Uppgift 18. Även solen har sina fläckar

Följande lilla problem i MATLAB kan lätt hamna i och bli förundrad över. Säg att vi vill rita upp roten till ekvationen $x^2 - 1 = 0$. Vi vet vid det här laget att det kan åstadkommas med *plot(roots([1,0,-1]), '*')*. Pröva det. Konstatera att resultatet är käpprätt åt skogen. Förklaringen kan du läsa dig till med *help plot* (andra stycket). Det egentliga skälet är att man givit *plot* olika betydelse beroende på vad första parametern är. Och det är uppenbarligen inte bra. Det här problemet dyker upp när samtliga rötter är reella. En nödlösning är att lägga till ett litet rent imaginärt tal (och hoppas att inte alla rötterna har en imaginärdel som är lika med -detta tal), dvs *plot(roots([1,0,-1])+0.001*i, '*')*. Förvissa dig om att detta fungerar.

Uppgift 19. Zooma

MATLAB har vissa interaktiva inslag som saknas i Maple (och vice versa för den delen). Rita upp kurvan från Uppgift 14b. Tag med *help zoom* reda på vad *zoom*-kommandot uträttar. Tag sedan med dess hjälp reda på ett av nollställena med litet bättre noggrannhet än vad den ursprungliga figuren tillåter.

*Uppgift 20. WWW-hjälp

Dokumentationen till MATLAB finns även som WWW-sidor (se sid 177). Tag den vägen reda på information om *ezplot*-kommandot.

Uppgift 21. Rotvandring med hemmagjord funktion

De flesta har nog inte hunnit med någon av de tidigare förekommande rotvandringsproblemen. Själva grundidén med denna typ av uppgift är att man vill visualisera hur ändringar i indata påverkar resultaten i fall där beroendet är komplicerat. T ex vilka rötter som påverkas mycket.

Mata in m-funktionen nedan (kommentarerna kan du ju strunta i) och kalla filen *animate_roots.m* (eller kopiera den från *~hallgren/Intro/*). Sätt dig även in i hur den fungerar.

```
function animate_roots(p,i,a)
    % p is a vector with the coefficients of a polynomial as for roots
    % i is an order, e g i=0 means the zero order coefficient
    % a is a vector with values for that coefficient
    % animate_roots will draw the roots for the different values of a
    % in a common window
    % the user has to press any key between the cases
    % E g animate_roots([4,3,2,1],1,2:0.1:3);
```

```
% will draw the roots when the first order coeff varies from 2 to 3
clf
hold off
for var=a
    p(length(p)-i)=var;
    r=roots(p);
    plot(r, '+');
    hold on;
    pause
end
hold off
```

Pröva den sedan för det fall som nämns bland kommentarerna och för fallet i Uppgift 8.

Uppgift 22. Extra-uppgift: Program med knappar och menyer och sånt

Tydligt kan man inom MATLABs ram skapa trevliga interaktiva program. Vi skall belysa den tekniken litet grand. Mata in följande som en m-fil `USER.m` (eller kopiera från den vanliga platsen). Förklaringar har jag förhoppningsvis presenterat under föreläsningen.

```
% This is a short m-file with commands only, they are executed
% when the name of the file (.m excluded) is written as a command

% Create a window for graphics
fig = figure;

% Create a button control
h1 = uicontrol('Position',[10,20,200,60]);
set(h1,'Callback','s=33; s+2');
set(h1,'String','Tryck här');

% Create another button control for ending the experiment
h2 = uicontrol('Position',[210,20,200,60]);
set(h2,'Callback','delete(fig)');
set(h2,'String','Avsluta');

% Create a scroll-bar control; the value is written
h3 = uicontrol('Position',[150,90,200,60]);
set(h3,'Callback','s=get(h3, 'value')');
set(h3,'Style','slider');
```

Anropa den med `USER`. Pröva de tre styrorganen. Notera att hissen ger ett värde s mellan 0 och 1. Modifiera sedan den så att kurvan $y = e^{sx} - s$ eller eventuellt kurvorna $y = e^{sx}$ och $y = x$ ritas upp på intervallet när man dragit i hissen. Använd alternativt verktyget *guide* för att ändra användarsnittet.

3 MATLAB på webben

Företaget som gör MATLAB har webbadressen www.mathworks.com. Där kan man hitta en del reklam för MATLAB. Man kan hitta mer nyttig information genom att söka efter MATLAB i YAHOOs webbkatalog på www.yahoo.com.

Laborationer i kursmomentet Datoranvändning E1

Laboration nr 6: Filformat för bilder

98-10-28 Thomas Hallgren

1 Introduktion

Denna laboration knyter an till föreläsning 5 i kursen, där jag pratade om hur olika typer av information kan representeras digitalt. Vi koncentrerar oss här på hur bilder representeras. Vilka filformat är lämpliga för olika typer av bilder? Hur bra kompression ger olika filformat? Vilken bildkvalité får man? Vi jämför formaten PPM, GIF, JPEG och PostScript. Som hjälpmedel använder programmen `xv`, `MATLAB` och `FrameMaker`.

2 Uppgifter

I denna laboration är det lite färre uppgifter än i de tidigare, så förhoppningsvis går det att hinna med allihop på två timmar. Skriv ner svaren på frågorna och visa för en handledare för att bli godkänd.

Uppgift 1. Titta på bilder med `xv`

Vi börjar med att titta på bilden `~hallgren/Intro/blomvas.ppm`. Som ändelsen `.ppm` antyder är bilden sparad i PPM-format. Detta är ett mycket enkelt format utan någon form av kompression. Hur stor är filen? _____

Starta programmet `xv` och titta på bilden. Man kan göra det på två sätt:

- Man kan ange namnet på bildfilen på kommandoraden när man startar `xv`.
- Man kan starta `xv` utan att ange en bild. Genom att klicka med höger musknapp i `xv`'s fönster kan man få fram en kontrollpanel där bland annat kan trycka på knappen `Load` för att välja en ny bild som ska visas.

I PPM-formatet lagras färgen på varje pixel i RGB-format (dvs man lagrar hur mycket av primärfärgerna rött, grönt och blått som ingår) i 3 bytes per pixel. Om man trycker på tangenten `i` i `xv`-fönstret visas ett informationsfönster, där man bland annat kan se hur många pixels bred och hög bilden är (resolution).

- Hur många pixlar bred och hög är bilden? _____
- Hur många pixlar blir det? _____
- Hur många bytes är det i PPM-filen utöver det som behövs för pixlarna? _____

(För den nyfikne: titta hur filen ser ut om man öppnar den med `nedit`. Läs manualbladet för PPM-formatet med kommandot `man ppm`.)

Uppgift 2. Vi undersöker JPEG-formatet

xv kan visa bilder lagrade i många olika format, och även spara i många olika format, så man kan alltså använda xv för att konvertera mellan bildformat. Vi ska utnyttja detta för att spara bilden från förra uppgiften i JPEG-format och GIF-format och jämföra resultaten.

Vi börjar med att prova JPEG-formatet. Tag fram xv:s kontrollpanel (om du inte redan gjort det). Tryck på knappen `Save`. Det dyker upp ett filvals-fönster. Gå till labbgruppens hemkatalog (om det inte redan är den katalogen som visas), välj JPEG i `Format`-menyn, kontrollera att det står `blomvas.jpg` filnamnsfältet längst ner (ändra om det behövs) och tryck på `OK`. Det dyker upp ytterligare ett fönster, där man kan välja vilken bildkvalité man vill ha, men låt de förvalda inställningarna vara kvar och tryck på bara på `OK`.

- Hur stor plats tar bilden när den lagras i JPEG-format? _____
- Hur många procent av den PPM-filens storlek är det? _____

Man undrar förstås nu vilken effekt detta har på bildkvalitén. JPEG är ju ett format som sparar plats genom att ta bort detaljer ur bilden baserat på vad som är lätt att se med blotta ögat. Vi får ta en titt på JPEG-bilden. I xv:s kontrollpanelfönster finns en ruta där namnen på använda filer visas. JPEG-bilden vi just sparade borde finnas med där. Dubbelklicka på den för att visa den i bildfönstret. Kontrollera bildkvaliten. Dubbelklicka på den ursprungliga PPM-bilden igen och jämför. Syns det någon stor/liten skillnad? _____

(För den nyfikne: man kan alltså välja bildkvalité när man sparar i JPEG-format. Prova att spara med högre/lägre kvalité och undersök vad detta får för effekt på filstorleken och bildkvalitén.)

Anm. När man jämför bilder kanske man vill se båda samtidigt. xv kan bara visa en bild i taget, men man kan ju starta två kopior av xv, om man vill...

Vi ska även undersöka vad JPEG-kompressionen har för effekt på en annan typ av bild.

- Öppna bilden `~hallgren/Intro/screendump1.ppm` och spara den i JPEG-format. Hur stor blev filen (i procent av PPM-filens storlek)? _____
- Syns det några skillnader mellan originalet och den komprimerade bilden i detta fall? _____
- Prova att titta på bilden med 2 gångers förstoring. (Välj alternativet `Double Size` i menyn `Image Size` i xv:s kontrollfönster. Notera vilken tangent man kan använda för att göra samma sak). Försök säga något om hur JPEG-kompressionen har påverkat bilden: _____

Uppgift 3. Vi undersöker GIF-formatet

Klicka fram den första bilden igen (`blomvas.ppm`) och spara den i GIF-format.

- Hur stor blev GIF-filen? _____

- Hur många procent av PPM-filens storlek är det? _____
- Hur många procent av JPEG-filens storlek är det? _____
- Jämför JPEG-bilden och GIF-bildens kvalitet. Är någon av dem betydligt bättre än den andra, och vilken i så fall? _____
- Vilket format är lämpligast för denna bild? _____

Klicka även fram den andra bilden (`screenump1.ppm`) och spara den i GIF-format.

- Hur stor blev GIF-filen? _____
- Hur många procent av PPM-filens storlek är det? _____
- Vilket format är lämpligast för denna bild? _____

Anm. Även GIF-formatet komprimerar bildinformationen, men till skillnad från JPEG-formatet används en förlustfri kompression. Färgen på varje pixel bevaras alltså exakt. Men GIF-formatet är begränsat till bilder med höst 256 olika färger (så att det räcker att lagra en byte per pixel). För enkla dator/handgjorda bilder, som `screenump1` ger GIF-formatet en exakt kopia av originalbilden. För mer färgrika bilder, som `tex` foton, måste man alltså på något sätt reducera antalet färger, vilket kan mer eller mindre märkbart försämra bilden. Bilden `blomvas` innehåller tydligen inte så många olika färger så GIF-versionen blev hyfsat bra, men det kan alltså bli betydligt sämre.

Uppgift 4. Vi undersöker PostScript-formatet

PPM, GIF och JPEG är exempel på pixelbaserade bildlagringsformat. Detta betyder att bilden delas in i punkter och färgen på varje punkt lagras. Ju finare indelning (dvs ju fler punkter) man använder, desto högre blir bildkvalitén, men storleken på filerna växer förstås också. Därför lagras man inte bilderna med fler punkter än nödvändigt för ändamålet. Typiska datorbildskärmar visar 75dpi (dots per inch, punkter per tum) eller lite mer.

PostScript är ett objektbaserat bildbeskrivningsformat, dvs man lagrar vilka grafiska objekt bilden är uppbyggd av, eller, om man så vill, hur bilden ritades. Detta passar bra för bilder som framställs på dator med något slags ritprogram, men förstås mindre bra för foton. Storleken på filerna beror på hur komplicerade bilder man lagrar, men är oberoende av storleken på bilden.

Anm. När PostScript utvecklades ville man ha ett utökningsbar standard för bildbeskrivning. Man löste detta genom att göra PostScript till ett programmeringsspråk! Detta ger förstås mycket stor frihet.

Vi ska nu jämföra en pixelbaserad bild som skapats för att visas på en bildskärm och en PostScript-version av samma bild, genom att rita en sinuskurva med MATLAB, importera den på två olika sätt i FrameMaker och skriva ut dem.

- Starta MATLAB och rita en jämn och fin sinuskurva som i Uppgift 3 i Laboration nr 5.
- Spara kurvan i PostScriptformat med MATLAB-kommandot

```
print sinuskurva.epsi -depsc2 -epsi
```

Den sista flaggan, `-epsi`, gör att en pixelbaserad kopia av bilden bakas in i PostScriptfilen. Detta gör att filen blir onödigt stor (i det här fallet 100KB istället för 7KB, eller något sånt). Man kan hoppa över detta, men då visas en grå ruta istället för bilden när man importerar den i FrameMaker. (Det blir rätt när man skriver ut dokumentet.)

- Gör en skärmavbildning av sinuskurvan. Man kan använda `xv` för detta genom att trycka på knappen `Grab` i kontrollpanelen. Det dyker då upp ett nytt fönster där det står att man kan avbilda ett fönster genom att klicka i det eller en godtycklig rektangel. Tryck på `Grab`-knappen i det fönstret och klicka sedan i fönstret med sinuskurvan. Spara bilden i GIF-format i filen `sinuskurva.gif`.

Anm. I Laboration nr 4 gjorde vi en skärmavbildning med hjälp av FrameMaker. Två fördelar med att göra det med `xv` är att man lätt kan avbilda ett helt fönster och att man kan spara i valfritt filformat, t ex GIF, istället för XWD-formatet, som är okomprimerat och därmed ger väldigt stora filer.

Vi har nu tillverkat de två versionerna av sinuskurvan, så starta FrameMaker och skapa ett nytt dokument i stående A4. Importera de båda bilderna och justera storleken på dem så att de blir ungefär lika stora och precis får plats på en sida.

- När man tittar på dokumentet på bildskärmen, är bildkvaliteten på någon av de båda sinuskurvorna betydligt bättre än den andra, och i så fall vilken? _____

Spara dokumentet. Skriv ut det.

- Undersök utskriften. Är bildkvaliteten på någon av de båda sinuskurvorna betydligt bättre än den andra, och i så fall vilken? _____

3 Avslutande kommentarer

Anledningen till att vi tittade på just JPEG och GIF är de är de två vanligaste formaten för bilder på webben. Ett nyare format, som utvecklades för att ersätta GIF är PNG. Det använder också förlustfri kompression, men är inte begränsat till 256 färger. Det finns även andra fördelar med PNG.

Anledningen till att vi tittade på PostScript var att illustrera att pixelbaserade bilder som är skapade för att se bra ut på bildskärmen inte är det bästa när man skriver ut på skrivare. Detta beror på att skrivare har högre upplösning, vanligen 300dpi eller 600dpi.

Mer information om olika bildformat kan man förstås hitta på webben. En bra sida att starta från är:

<http://www.dcs.ed.ac.uk/~mrx/gfx/>

Laborationer i kursmomentet Datoranvändning E1

Laboration nr 7: Mer om FrameMaker och lite annat

1996, 1997 Magnus Bondesson
98-11-05 Thomas Hallgren

1 Introduktion

I denna laboration gör vi först några nya övningar i FrameMaker och Netscape. Sedan följer några övningar som tar upp komprimering av filer. Sist finns några övningar på UNIX-kommandon och några finurliga sätt att sätta ihop existerande kommandon till nya användbara kommandon.

Som förberedelse inför denna laboration kan det vara lämpligt att läsa de avsnitt i Gula Boken som det finns hänvisningar till i de olika uppgifterna.

2 Uppgifter

Som tidigare markerar symbolen en konkret arbetsuppgift.

Skriv ner svaren på frågorna och visa för en handledare för att bli godkänd. Handledaren vill förmodligen också titta på FrameMaker-dokument och andra filer du skapat under arbetets gång.

2.1 FrameMaker

I Laboration nr 4 ägnade vi oss åt FrameMaker och du producerade ett dokument baserat på mallen *Rapport normal*. Nu jobbar vi vidare lite på detta, så ta fram ditt dokument och spara det under ett nytt namn, tex `Rapport7.doc`, innan du går vidare. (När senare uppgifter refererar till `Rapport7` är det ditt dokument som menas.)

Uppgift 1. Egen styckeutformning

Egna utformningar tas upp i Gula Boken i avsnitt 7.7.

Det händer då och då att man vill ha en styckeutformning som inte finns färdig i styckekatalogen. Din uppgift här är att skapa en sådan och placera den i styckekatalogen. Den nya utformningen skall heta *Citat*, ha kursiv stil och vara indragen 3 cm i förhållande till vänstermarginalen. Använd menyn **Utforma/Stycken/Utformning** för att få tillverkningsdialogen och sedan den dialogens **Kommandon/Ny utformning** för att få resultatet att hamna i styckekatalogen. Prova sedan att utformningen fungerar som den skall.

Uppgift 2. Egen teckenutformning

- Att göra en teckenutformning och placera den i teckenkatalogen går till på ungefär samma sätt. I teckenkatalogen för ditt dokument finns antagligen *Betoning* som kursiverar markerade tecken. Gör en ny teckenutformning för nyckelord, och låt dem visas med röd text och fetstil. Kalla den nya teckenstilen *Nyckelord* och lägg den i teckenkatalogen. Använd menyn **Utforma/Tecken/Utformning** för att få tillverkningsdialogen och sedan den dialogens **Kommandon/Ny utformning** för att få resultatet att hamna i teckenkatalogen.

Uppgift 3. Ändra stil på hela dokumentet

Om man har utformat sitt dokument genom att konsekvent använda de namngivna utformningarna i stycke- och teckenutformningskatalogerna kan man ganska lätt ändra stil på hela dokumentet genom att importera utformningar från ett annat dokument.

Tillvägagångssätt: man väljer **Arkiv/Importera/Utformningar...** från menyn. I fönstret som visas finns en meny där man kan välja vilket dokument man vill importera från. Här visas andra dokument som är öppna, så man måste alltså öppna det dokument man vill importera utformningar från först. Nyskapade dokument visas inte förrän efter man har sparat dem första gången. Det finns också en mängd knappar för att ställa in vad som ska importeras. Alla alternativ är påslagna från början och det är väl lämpligt om man vill man ändra stil helt och hållet på sitt dokument.

- Prova att importera utformningarna från standardmallen *RapportNumerisk* eller *RapportKantrubruk* i ditt eget dokument. (För att se finessen med den senare bör man ha några stycken med utformningen *Rubrik2* i sitt dokument.)

Uppgift 4. Numrerade figurer

I rapporter är det vanligt att man numrerar figurer och hänvisar till dem från ett eller flera ställen i den löpande texten.

- Det finns inget färdigt verktyg för att skapa numrerade figurer i FrameMaker. I dokumentmallen *Rapport Normal* finns det en styckeutformning som heter *Bild*, som det är tänkt att man ska använda när man skapar en numrerad figur. Själva figuren gör man plats för genom att sätta in en kopplad ram i stycket. Ramens placering ska vara **Under aktuell rad**. Prova att göra en figur på detta sätt.

En vanlig stil när det gäller placering av figurer är att de läggs under den rad där man refererar till figuren, om det finns plats. Annars låter man den löpande texten fortsätta och lägger figuren överst på nästa sida. Denna metod får man om man kryssar för alternativet **Flytande** när man skapar kopplade ramar i FrameMaker.

- Nackdelen med ovanstående metod för numrering av bilder är att bildtexten inte följer med figuren om figuren hamnar på något annat ställe än under aktuell rad. Prova t ex att markera ramen du nyss gjorde och välj placeringen **Överst i spalten**.

- Ett annat sätt numrera figurer är att använda tabeller istället för kopplade ramar. Dokumentet `~hallgren/Intro/Figurer.doc` innehåller ett exempel på hur man kan göra. Öppna det och ta en titt.

- Importera stycke- och tabellutformningarna från `Figurer.doc` i ditt eget dokument och skapa en numrerad figur.

Uppgift 5. Generera PostScript-filer

Hur gör man om man vill göra sina FrameMaker-producerade dokument tillgängliga för andra via webben? Ett sätt är förstås att kopiera FrameMaker-filen till sin webbplats, men då måste den som vill titta på dokumentet ha FrameMaker, och det är det ju långt ifrån alla som har. Ett bättre alternativ är att skapa en PostScript-fil.

PostScript är det format FrameMaker använder för utskrift på skrivare, så att skapa en PostScript-fil är en variant på att göra en utskrift. Man väljer alltså **Skriv ut...** i **Arkiv**-menyn och kryssar i rutan **Skriv till fil**. FrameMaker föreslår ett filnamn, men man kan ändra på det om man vill.

- Skapa en PostScript-fil för ditt dokument.

Uppgift 6. Generera PDF-filer

Nyare versioner av FrameMaker (t ex 5.5) kan generera filer i PDF (Portable Document Format), en variant på PostScript som har blivit populär för elektronisk spridning av dokument. För att titta på PDF-filer kan man använda Acrobat Reader, som är gratis och finns för flera olika datorsystem.

Den version av FrameMaker som finns på E:s datorer är 5.1, så tyvärr kan vi inte öva på att skapa PDF-filer, men vi kan titta på ett färdigt PDF-dokument för att se hur det kan se ut.

- Ta en titt på PDF-filen `~hallgren/Intro/Laborationer.pdf`. Unix-kommandot för att starta Acrobat Reader heter `acroread`. Notera att man snabbt kan hoppa till olika delar i dokumentet genom att klicka i listan till vänster och att man kan klicka på trianglarna i listan för att ändra listans detaljnivå. Prova också att klicka på korsreferenser i texten.

2.2 Netscape

Uppgift 7. Surfa lokalt

Den vanligaste användningen av Netscape är väl att surfa på webben, men man kan faktiskt surfa bland sina egna filer också. Vi tittade på lokala filer redan i Laboration nr 1. Faktum är att man även titta på kataloger. Filer och underkataloger blir då länkar, så istället för att använda `cd`, `ls`, `more` och andra kommandon för att bläddra runt och titta på filer kan man enkelt klicka sig fram med Netscape.

- Titta på labbkontots hemkatalog med Netscape genom att skriva in `file:` följt av gruppens hemkatalog i **Location**-rutan (Vilket det är kan man ta reda på med kommandot `pwd`. `~`-notationen fungerar inte, tyvärr). Passa på att sätta ett bokmärke, om du lätt vill kunna göra detta fler gånger.

Anm. Du märker säkert att Netscape visar filnamn som börjar med `.`, vilka i normala fall inte brukar visas.

- I katalogen finns förmodligen diverse olika slags filer. Netscape känner förstås inte till *alla* filformat. HTML-filer visas förstås i Netscapes fönster, men vad händer om man klickar på
 - a. en enkel textfil (t ex Berling.txt)? _____
 - b. en GIF-bild? _____
 - c. en JPEG-bild? _____
 - d. en PostScript-fil? _____
 - e. en m-fil från MATLAB? _____
 - f. ett FrameMaker-dokument? _____

Uppgift 8. Ändra Netscapes inställningar

Som du upptäckte i förra uppgiften startar Netscape ibland andra program för att visa filer av typer som Netscape inte kan visa själv. Det finns inställningar, som man kan ändra på om man vill, som avgör hur olika filtyper hanteras. I Netscape version 3 hittar man dessa genom att i menyn **Options** välja alternativet **General Preferences...** och i fönstret som dyker upp klicka på **Helpers**. Då ser man en lista med filtyper.

Vi provar att ändra inställningen för hur GIF-bilder hanteras.

- Leta upp filtypen `image/gif` i listan. Klicka på den och sedan på knappen **Edit...** Det dyker upp ett fönster med inställningar för GIF-filer. I rutan **Handle By** kan man välja bland annat låta Netscape själv visa filen (alternativet **Navigator**) eller att starta ett annat program för att visa filen (alternativet **Application**). Förmodligen är det senare alternativet valt (dvs GIF-bilder visas av något annat program), så prova att ändra så att Netscape själv visar GIF-bilder. Om det redan var den inställningen som var vald, prova att välja **Application** och skriva in kommandot `xv %s` i rutan. (När Netscape sedan startar `xv` kommer `%s` att ersättas med ett filnamn.) Tryck på **OK** (i två fönster) och prova att klicka på någon GIF-bild i din hemkatalog och kolla att det fungerar.
- (Om du får tid över) Om man tycker att texten på webbsidorna i Netscape är svår-läst, för stor eller för liten kan man ändra inställningarna under **Options/General Preferences.../Fonts**. Personligen tycker jag att New Century Schoolbook är ett trevligare typsnitt än Times, som brukar vara det förvala.

2.3 Komprimering

När man lagrar information kan man spara plats genom olika former av komprimering. I Laboration nr 6 tittade vi på olika filformat för bilder, varav några (GIF & JPEG) hade inbyggd komprimering. Det finns även program som kan komprimera godtyckliga filer. `gzip` är ett ofta använt sådant i Unix (i Windows är WinZip ett populärt program. På Macintosh finns t ex StuffIt och Compact Pro).

Generella komprimeringsprogram fungerar alltså på godtyckliga filer. Kan man verkligen komprimera information utan att veta något om dess uppbyggnad? Vi tar reda på det genom att göra några komprimeringsexperiment. `gzip` används enligt:

```
gzip filnamn
```

där *filnamn* är namnet på den fil man vill komprimera. Denna ersätts då med en komprimerad fil som får tillägget `.gz` sist i namnet. Man kan dekomprimera en fil med kommandot

```
gunzip filnamn.gz
```

där *filnamn.gz* är namnet på den komprimerade filen. Filen ersätts då med en dekomprimerad fil med `.gz` borttaget ur namnet. Innehållet i denna är exakt samma i den ursprungliga okomprimerade filen, dvs komprimeringen är förlustfri.

Uppgift 9. Komprimeringsexperiment

- Prova `gzip` på några filer och fyll i tabellen (om du inte redan har filerna kan du kopiera dem från `~hallgren/Intro/`):

Fil	Ursprunglig storlek (s0)	Komprimerad storlek (s1)	Kompressionsfaktor (s0/s1)
Berling.txt			
Propaganda.txt			
blomvas.ppm			
screendump1.ppm			
blomvas.jpg			
blomvas.gif			
lab_7.doc			
lab_7.ps			
lab_7.pdf			

- Klarar Netscape av att visa en komprimerad fil? Prova att klicka på t ex den komprimerade versionen av `Propaganda.txt` (dvs `Propaganda.txt.gz`). Vad händer? _____

- Med utgångspunkt från dessa experiment kanske du kan våga dig på att säga något om vad de filer som går bra att komprimera på detta sätt har gemensamt?

2.4 UNIX-kommandon, omdirigering och rörledning

Uppgift 10. Några användbara UNIX-kommandon: Folk och datorer

- Pröva följande tre kommandon. Se manualblad eller Gula Boken för mer information.
- `finger användarnamn` eller `förnamn` eller `efternamn`
Ger information om alla användare med aktuellt namn
 - `rwho`
Ger information om alla påloggade användare vid E-systemet
 - `v`
Lokalt UNIX-kommando, som ger information om lediga/upptagna datorer.

Uppgift 11. Omdirigering

Normalt hamnar resultaten av ett UNIX-kommando i det fönster i vilket kommandot givits, vilket brukar kallas "standard-ut". Men man kan också dirigera resultaten så att de i stället hamnar i en vanlig fil, med (`>` kan utläsas "från kommandot till")

```
kommando >filnamn
```

Läs mera om omdirigering i avsnitt 1.16.1 i Gula Boken.

- Se till att resultatet av `ls -l` hamnar som filen `LISTA`. Kontrollera med `tex more` att `LISTA` innehåller det den skall.
- Programmet `~hallgren/Intro/slumpa` genererar en massa slumpstal. Kör programmet. Kör det sedan så att resultaten i stället hamnar som filen `slumptal`.

På motsvarande sätt kan ett kommando som normalt hämtar information från tangentbordet ta den från en fil. Man använder då i stället tecknet `<` ("till kommandot från"). Vi avstår från någon uppgift på detta.

Uppgift 12. Några användbara UNIX-kommandon: Filer

Låt oss titta på tre kommandon `wc`, `grep` och `sort` som alla kan användas på två olika sätt. Antingen skriver man ett filnamn tillsammans med kommandot eller också skriver man in information på raderna efter kommandot och anger indataslut med CTRL-D. Mer information på vanligt sätt (manualblad resp sid 58).

- Tag med hjälp av `wc` reda på antal rader, ord och tecken i filen `Berling.txt`.
- Kommandot `grep` används för att söka efter teckenföljder i en fil. När `grep` funnit en rad som innehåller den sökta textsträngen så skriver `grep` ut denna rad. Skriv ut de rader i filen `Berling.txt` som innehåller teckenföljden `hon`. Ta också reda på vilka som innehåller ordet `hon`. (Tips: läs vad `grep -w` betyder i manualbladet för `grep`.)
- Kommandot `sort` används för att sortera text eller tal. Sortera raderna i `Berling.txt` alfabetiskt.

Uppgift 13. Rörledningar (pipes)

Läs om hopkoppling av kommandon i Gula Boken, sid 23.

- Ange hur man med hjälp av hopkoppling av kommandon på ett smidigt och enkelt sätt kan se hur många hemkataloger det finns på datorsystemet som tillhör E-teknologer inskrivna 1998 (katalogen /u1/e98; använd kommandona `ls` och `wc`): _____
- Hur många kataloger finns det? _____
- Det tidigare nämnda programmet `slumpa` producerar osorterade slumpstal. Skriv en rörledning som gör att de kommer ut sorterade i växande ordning på skärmen (*sort* behöver kanske en extra flagga).

Uppgift 14. Kommandofiler

När man startar ett terminalfönster startas normal kommandotolken `tcsh`, som läser de kommandon man matar in på tangentbordet och utför dem. En naturlig fråga är nu: kan `tcsh`, liksom kommandona vi tittade på i Uppgift 12, även läsa indata från en fil? Svaret är ja! Har man en fil som innehåller kommandon kan man få `tcsh` att utföra dem genom att skriva `tcsh filnamn`.

Filer som innehåller kommandon kallas inte helt oväntat för **kommandofiler** (se sidan 29 i Gula Boken). En kommandofil är alltså en textfil (tillverkas med t ex *NEdit*) bestående av ett antal kommandorader.

- Gör med *NEdit* en fil med innehållet

```
date
echo $1 $USER
```

Spara den under namnet `kommando1`. Provkör kommandofilen genom att mata in `tex`

```
tcsh kommando1 Hejsan
```

Vad skrivs ut? _____

Förklaring: `$1` är en sorts variabel som ersätts med det ord som skrivs efter `kommando1` och `$USER` ersätts med användarnamnet. Kommandot `echo` skriver ut en textföljd, dvs kommandofilanropet kommer att skriva ut datumuppgifter och sedan `Hejsan användarnamnet`.

Det går även att göra kommandofiler lika lätta att använda som vanliga kommandon (dvs man slipper skriva namnet på kommandotolk varje gång man använder dem). `kommando1` görs körbar med UNIX-kommandot `chmod` enligt

```
chmod +x kommando1.
```

Därefter används kommandofilen med `tex`

```
kommando1 Hejsan
```

- Prova detta!

Anm. Kommandofiler som används på detta sätt tolkas av kommandotolken `sh`, inte `tcsh`, som på E är den normala kommandotolken för det man själv skriver in i ett terminalfönster. I grunden fungerar båda på samma sätt, men lite mer avancerade saker skrivs på olika sätt. Vill man ange explicit vilken kommandotolk en kommandofil ska tolkas av kan man ange detta på första raden i filen. För att få `tcsh` kan man skriva

```
#!/usr/local/pd/bin/tcsh -f
```

Uppgift 15. e8rwho



Tillverka ett nytt kommando som fungerar som `rwho`, men utskriften begränsas till användare vars namn börjar på `e8`. Kalla kommandot `e8rwho`.

3 Avslutande kommentarer

I år är det meningen att alla projektgrupper ska göra en hemsida på webben för sina projekt och göra projektrapporten tillgängliga via den. Lämpliga format att använda för rapporten är då HTML, förstås, eller PDF. Version 5.5 av FrameMaker har stöd för generering av såväl HTML som PDF. Tyvärr har E-sektionen bara version 5.1, så vi kommer förmodligen att få nöja oss med att generera PostScript-versioner av rapporterna.

Kommentarer till FrameMaker: man kan lära sig mycket genom att själv experimentera och utforska möjligheterna. Det vi hinner ta upp allt på laborationerna är ju bara ett litet smakprov.

Kommentar till att använda Netscape för att bläddra bland egna filer: de som tidigare har använt Macintosh eller Windows på PC är förmodligen vana att enkelt kunna bläddra runt bland sina filer (med Finder respektive Utforskaren) och tycker kanske det verkar fånigt att använda Netscape för detta. Faktum är att det finns program för Unix som presenterar filer och kataloger ungefär som på Mac/Windows, men inget sådant finns installerat på E:s datorer, så vitt jag vet...

3.1 Information på webben

Det är företaget Adobe som gör både FrameMaker och Acrobat Reader. Adobe har webbsidor på `www.adobe.se` och `www.adobe.com`. Man kan hitta information om de två programmen på textsidorna

```
http://www.adobe.com/prodindex/acrobat/
```

```
http://www.adobe.se/produktinfo/framemaker/
```

Information om Netscape 3.0 finns på (fås också med **Help/Handbook** i menyn):

```
http://home.netscape.com/eng/mozilla/3.0/handbook/
```

Komprimeringsprogrammet `gzip` har en egen webbadress: `www.gzip.org`! Man kan hitta information komprimering och fler komprimeringsprogram i Yahoo-katalogen:

```
http://dir.yahoo.com/Computers_and_Internet/Software/  
System_Uilities/Compression/
```

Laborationer i kursmomentet Datoranvändning E1

Laboration nr 8: Nätverk och prestanda

98-11-22 Thomas Hallgren

1 Introduktion

På de flesta ställen där det finns mer än en dator är datorerna hopkopplade i ett lokalt nätverk. UNIX erbjuder ett sätt att använda nätverk som inte brukar förekomma i PC/Macintosh-nätverk, nämligen möjligheten att köra program på andra datorer än den man själv sitter vid (på engelska *remote execution*). I denna laboration utnyttjar vi denna möjlighet för att enkelt köra ett benchmark (ett program som testar hur snabbt en dator löser en viss uppgift) på flera olika datorer. Övningarna belyser också att fönstersystemet X Windows (dvs det fönstersystem som är standard i UNIX) är *nätverkstransparent*, dvs vilken dator programmet körs på och vilken bildskärm programmets fönster visas på kan väljas oberoende av varandra, samt att det inte alltid lönar sig att köra ett program på en annan dator, även om den är snabbare.

Kommandon och program som dyker upp i denna laboration är: `rlogin`, `slogin`, `logout`, `ps`, `ping`, `traceroute`, `xmfract` och `xaos`.

2 Uppgifter

Som tidigare markerar symbolen en konkret arbetsuppgift.

Skriv ner svaren på frågorna och visa dem för en handledare för att bli godkänd.

2.1 Uppgifter från förra veckan

Laboration nr 7 tog lite mer än två timmar att göra för de flesta, så om du inte hann göra klart alla uppgifterna, gör det nu.

2.2 Att köra program på andra datorer

Några vanliga tjänster som lokala nätverk brukar erbjuda är:

- skrivardelning (man kan ha en gemensam skrivare, istället för en per dator).
- fildelning (filer som finns på hårddisken i en dator är tillgängliga på andra datorer).

I UNIX-system brukar detta vara gjort så att man inte märker att nätverket finns. Alla datorerna verkar likadana, dvs alla filer, program och skrivare är tillgängliga på alla datorerna, utan att man behöver veta vilken dator de egentligen finns på.

Anm. Att det är så är ju väldigt bekvämt. Det är antagligen en kvarleva från tiden då persondatorer, arbetsstationer och nätverk inte fanns och man istället bara hade en

enda, stor och klumpig dator med flera textterminaler inkopplade via långa sladdar. Då fanns alla filer på samma dator och alla terminalerna fungerade likadant.

I typiska nätverk med UNIX-datorer spelar det alltså ingen större roll vilken dator man sitter och arbetar vid, när det gäller vad man har tillgång till. Datorerna kan dock skilja sig på ett antal punkter:

- De kan ha olika bra bildskärmar, tangentbord och mus.
- De kan ha olika prestanda, t ex vara olika snabba och ha olika mycket primärminne.

På många arbetsplatser står det nuförtiden en dator på varje skrivbord. Ett sätt att hålla nere kostnaderna, samtidigt som man ger alla användare tillgång till en kraftfull dator för krävande beräkningar, är att ha enkla, billiga datorer på skrivborden och en gemensam snabb, men dyr dator åtkomlig via *remote execution*. Detta fungerar ofta bra, eftersom det typiska är att en användare bara utnyttjar datorns kapacitet en liten bråkdel av tiden.

Anm. Detta sätt att bygga upp datormiljön påminner lite om det gamla sättet, dvs en enda dator som används via enkla textterminaler.

I UNIX-system brukar det vara väldigt enkelt köra program på vilken dator som helst i det lokala nätverket. Man använder sig av något av kommandona `rsh`, `ssh`, `telnet`, `rlogin` eller `slogin`. De två första är lämpliga för att köra ett enstaka kommando, de tre följande är lämpliga när man ska köra flera kommandon, eller interaktiva program. `ssh` (secure shell) och `slogin` (secure login) är efterföljare till `rsh` (remote shell) respektive `rlogin` (remote login), som har UNIX-bakgrund. `telnet` är ett mera systemoberoende kommando.

Uppgift 1. Vi testar några datorer med Dhystone-testet

Hur snabba är de olika datorerna på E-sektionen? Vi provar det genom att köra Dhystone-testet, som är ett gammalt enkelt benchmark (dvs program för att jämföra prestanda), som mäter hur snabb en dator är på heltalsberäkningar och enkel texthantering.

Dhystone-programmet finns i filen `~hallgren/Intro/dhry`.

Programmet frågar hur många gånger man vill att testet ska köras. På våra datorer är 1000000-2000000 lagom. (När programmet skrevs räckte det kanske med 10000 gånger.) Tar man för få klagar programmet på att mättiden blir kort för att resultatet ska vara pålitligt.

Provkör `dhry` på den dator du sitter vid. Vilken dator är det? _____

När testet är klart skrivs en del kontrolltexter ut. Sist kommer resultatet, antalet Dhystone per sekund. (Programmet mäter själv tiden.) Vad blev resultatet för den dator du sitter vid? _____

Provkör programmet `dhry` på några fler datorer och fyll i Tabell 1. Använd kommandot `rlogin datornamn` för att "logga in" på de olika datorerna. Avsluta med kommandot `logout` när du är klar med varje dator.

Dator	Dhrystones per sekund
quarl32	
hortensia	
quarl102	
quarl202	

Tabell 1: Resultat av Dhrystone-testet på några av E-sektions datorer

Anm. En detalj att lägga märke till när man använder `rlogin` är att man inte behöver tala om vem man är innan man släpps in på den andra datorn. Detta beror på att `rlogin` automatiskt skickar över ditt användarnamn till den andra datorn och att datorerna i det lokala nätverket litar på varandra.

2.3 Att köra fönsterprogram på andra datorer

Vi har nu sett att program som körs i ett terminalfönster lätt kan köras på en annan dator än den man sitter vid. Hur går det med program som öppnar egna fönster? Fönstersystemet i UNIX, X Windows, är *nätverkstransparent*, dvs det spelar ingen roll om bildskärm, mus och tangentbord (som tillsammans utgör en *X-terminal*) finns på samma dator som programmet körs på, eller om det är ett nätverk mellan dem.

Valet av X-terminal styrs av omgivningsvariabeln `DISPLAY`. Man kan använda kommandot `echo $DISPLAY` för att se hur den är definierad och `setenv DISPLAY ...` för att definiera om den. (Vi har tidigare stött på omgivningsvariabeln `PRINTER`, som på samma sätt styr valet av skrivare). Det vanliga är dock att man vill ha fönsterna på den dator man sitter vid, så man behöver inte bry sig så mycket om detta.

När man använder `rlogin` eller `telnet` för att logga in på en annan dator blir inte `DISPLAY`-variabeln definierad automatiskt. Det kan också av säkerhetskäl hända att det inte räcker att ställa in den rätt heller. Kommandot `slogin` löser dessa båda problem på ett smidigt sätt, så vill man köra fönsterprogram på en annan dator är det lämpligt att logga in med `slogin`.

`ssh` och `slogin` har ett säkerhetssystem som baseras på kryptering med `s k` öppna nycklar. Sådana skapas första gången man använder något av kommandona och ett meddelande som upplyser om detta dyker upp. Det dyker också upp en fråga första gången man kopplar sig till en viss dator.

Uppgift 2. Kör fönsterprogram på olika datorer

Programmet `xmfract`, som visar fraktaler, finns i `~hallgren/Intro/`. Vi provar att köra det på den långsammaste och den snabbaste datorn enligt Tabell 1.



- Ta fram ett terminalfönster och logga in med `slogin` på den långsammaste datorn. Starta programmet `xmfract` i bakgrunden (genom att skriva ett `&`-tecken sist på kommandoraden). Klicka bort eventuella meddelanderutor som visas.

- Tag fram ett nytt terminalfönster och logga in på den snabbaste datorn. Starta `xmfract` på samma sätt som ovan.

Välj menykommandot **Image/Create Image** i båda `xmfract`-fönstren. Programmen ritlar nu upp en bild av Mandelbrot-mängden. Förhoppningsvis märker man nu en viss skillnad i hastighet på de båda datorerna.

Kommandot `ps` visar information om de program man har startat. Kör det båda terminalfönstren och leta upp `xmfract` i listan. Kolumnen `TIME` visar hur mycket processortid programmet har använt.

- Hur mycket processortid behövdes för att generera bilden på den långsamma datorn? _____
- Hur mycket tid behövdes på den snabba datorn? _____
- Stämmer skillnaden i tid med hur snabbt bilden ritades enligt din uppfattning? ____

Det är ju inte bara hur snabb processorn är som avgör hur lång tid det tar att utföra en viss uppgift. Om du svarat ja på ovanstående fråga är det förstås processorn som har störst inverkan i det här experimentet. Om du svarat nej är det andra faktorer. En sådan faktor kan vara nätverkets prestanda.

2.4 Att undersöka nätverkets prestanda

När lönar det sig att köra ett program på en annan dator som är snabbare än den man sitter vid? Program kan delas in i *CPU-intensiva* och *IO-intensiva*, dvs program vars körning domineras av beräkningar respektive in- och utmatning. Det finns förstås också mellanting. Ett program som är CPU-intensivt, som t ex Dhrystone-testet, vinner förstås på att köras på en snabbare dator. Ett program som är IO-intensivt, som t ex ritlar mycket i sitt fönster, visar stora bilder eller film, kanske blir långsammare om man kör det på en annan dator. Nätverkets prestanda har då större betydelse en datorns.

Två parametrar bestämmer ett nätverks prestanda. Den första är fördröjningen: hur lång tid tar det för ett meddelande som skickas genom nätverket att komma fram? Den andra är kapaciteten: hur mycket data kan överföras på en viss tid?

Uppgift 3. Mät fördröjningar med ping

Kommandot `ping datornamn` skickar ett meddelande till en dator, denna skickar genast (förhoppningsvis) ett svar. Detta gör att man kan mäta fördröjningarna i nätverket.



Undersök hur stor fördröjningen är vid kommunikation mellan den dator du sitter vid och några andra datorer och fyll i kolumnen Ping-tid i Tabell 2. Kommandot `ping` skickar ett meddelande i sekunden till man avbryter det med `CTRL-C`. Man kan också begränsa antalet meddelanden t ex till 2 genom att skriva `ping -c 2 datornamn`.

Kategori	Datornamn	Ping-tid	Hopp
En annan dator på E	hortensia		
En dator någon annanstans på Chalmers	dogbert.cs.chalmers.se		
En dator i norra Sverige	www.luth.se		
En dator i USA	www.yahoo.com		
En dator i Japan	www.yahoo.co.jp		

Tabell 2: Mätning av fördröjningar och avstånd i nätverket

Uppgift 4. Undersök nätverkets kapacitet

Istället för att försöka mäta hur stor överföringskapacitet nätverket på E har gör vi ett praktiskt prov.

Program som visar rörliga bilder skickar stora mängder data till sitt fönster. Ett räkneexempel: hur många megabyte per sekund behöver överföras då 20 bilder per sekund visas i ett fönster som är 480×360 pixels stort (med 1 byte per pixel)? _____

- Programmet `xaos`, som också det finns i `~hallgren/Intro/`, ritar mandelbrot-mängden så snabbt att man kan zooma i den i realtid. Starta `xaos` på den dator du sitter vid *och* på en annan, minst lika snabb dator. Prova att zooma i de båda fönstren genom att hålla vänster musknapp nertryckt (man zoomar ut med höger musknapp). Du märker förhoppningsvis en viss skillnad mellan fönstren. Beskriv skillnaden:

Lönar det sig att köra detta program på en annan, snabbare dator? _____

Uppgift 5. Vägar genom Internet

- Internet kopplar ihop lokala nätverk på olika platser i världen. Meddelanden som skickas via Internet passerar normalt många kopplingspunkter (routers) på vägen. Kommandot `traceroute datornamn` visar vilken väg genom nätet meddelanden till en viss dator skickas. Kör det och undersök hur många hopp meddelanden gör (hur många ställen det landar på, inklusive målet) när de skickas till datorerna i Tabell 2.

3 Avslutande kommentarer

Remote execution tillåter att man använder en kraftfull dator utan att man behöver vara i närheten av den. Man t ex sitta vid en dator på E-sektionen och köra simuleringar på en supersnabb paralleldator på andra sidan jordklotet.

Innan man startar ett program på en annan dator kan det förstås vara lämpligt att kontrollera om någon annan redan använder datorn. Några användbara kommandon är:

- `v` - visar vilka datorer som är lediga (finns endast på E-sektionen).

- `who` - visar vilka som är inloggade på den dator kommandot körs på.
- `w` - visar vilka program användare på datorn kör. (`ps` kan fås att visa mer.)
- `uptime` - visar medelvärden över hur hårt belastad datorn har varit den senaste tiden. 0.00 = datorn gör inget, 0.50 datorn har arbetat med något 50% av tiden, 1.00 = datorn är upptagen med något hela tiden, 2.00 = datorn har i genomsnitt två saker olika saker att göra och växlar mellan dem, o s v.

Kom ihåg att `man`-kommandot ger detaljerade beskrivningar av kommandon.

3.1 Vad händer när man kör två program samtidigt på en dator?

Hur bra det går att köra två program samtidigt på en dator beror på vilka resurser i datorn de använder. Man kan i princip köra ett CPU-intensivt program samtidigt med ett IO-intensivt program samtidigt utan att något av dem störs av det andra.

Kör man flera CPU-intensiva program samtidigt fördelas processorns tid rättvist mellan alla programmen. Processorn kan tämligen snabbt växla mellan olika beräkningar, så det går ungefär lika fort att köra två program samtidigt som att köra det ena efter det andra.

Hårddiskar är däremot inte så snabba på att växla mellan att läsa/skriva på olika ställen på disken, så om man kör två program som båda läser/skriver stora filer kan det gå betydligt långsammare om man kör dem samtidigt istället för ett i taget.

Det blir också problem om program som körs samtidigt tillsammans behöver mer minne än det finns i datorn. Tack vare det virtuella minnessystemet kan programmen fortsätta köra, men data kommer att kopieras fram och tillbaka mellan primärminne och den s k swaparean på hårddisken, vilket tar mycket tid, och datorn kan bli väldigt långsam.

3.2 Information på webben

På CPU Info center, <http://infopad.eecs.berkeley.edu/CIC/>, hittar man massor med information om olika processorer, bland annat en lista med benchmarkresultat.

SPEC, The Standard Performance Evaluation Corporation, finns på www.spec.org. De presenterar bland annat resultaten av sina benchmark (SPECint95, SPECfp95, m fl).

På <http://www.top500.org/top500.list.html> finns en lista på världens 500 snabbaste datorer. Två av dem finns på Chalmers.

På <http://www.internettrafficreport.com/> visas med olika diagram hur hårt Internet är belastat.

Information om `ssh` (och `slogin`) hittar man om man går till www.yahoo.com och fyller i `ssh` i sökrutan. Det finns gratisversioner för Unix och Windows.

En lathund över kommandon som använts i laborationerna kommer i veckan att dyka upp på kursens hemsida.

Laborationer i kursmomentet Datoranvändning E1

<http://www.etek.chalmers.se/~hallgren/Eda/>

Laboration nr 9: Mer om WWW och HTML

1996, 1997 Magnus Bondesson
98-11-15 Thomas Hallgren

1 Introduktion


Denna laboration handlar om diverse saker som man kan ha nytta av när man skapar webbsidor. Det ingår också en övning på att hämta hem ett program från Internet, kompilera det och använda det. Det blir också lite repetition av saker som dykt upp tidigare i kursen.

Kommandon och program som dyker upp i denna laboration är: `gnnpres`, `xpaint`, `xv`, `giftrans`, `vrweb`, `tar`, `make`, `whirlgif` och `xanim`.

1.1 Förberedelser

Det kan vara lämpligt att titta igenom gamla laborationer, speciellt med tanke på Uppgift 3.

1.2 Redovisning

Denna laboration redovisas genom att man skapar en webbsida som innehåller svaren från uppgifterna. Saker som ska tas med i redovisningen är markerade med  .

För att bli godkänd på denna laboration ska du visa upp redovisningssidan för en handledare.

1.3 HTML och verktyg för att skapa HTML-dokument

Som vi såg redan i Laboration nr 1 så skrivs webbsidor i beskrivningsspråket HTML. Många ordbehandlingsprogram (men tyvärr inte den version av FrameMaker som finns på E) kan nuförtiden spara dokumenten i HTML-format.

HTML är dock ett ganska enkelt format, så man kan också använda vanliga textredigeringsprogram, t ex `nedit`. En ganska lättläst HTML-guide finns på

<http://www.htmlhelp.com/reference/wilbur/>

och man kan också lära sig mycket på att titta på HTML-koden för existerande webbsidor, t ex genom att använda menyalternativet **View/Document Source** i Netscape. När man hittar något bra på en webbsida som man inte riktigt förstår kan man gå till HTML-guiden och slå upp det.

Det finns också program speciellt gjorda för att skapa webbsidor. Ett sådant är *GNNpress* och det finns på E.

I denna laboration kan du i de flesta uppgifterna välja om du vill använda *GNNPress* eller något annat verktyg för att skapa webbsidor, eller om du vill skriva HTML-koden själv i ett textredigeringsprogram.

GNNpress startas med kommandot `gnnpres`. Det används på ungefär på samma sätt som vanliga ordbehandlingsprogram. Men allt är ändå inte självklart, så fråga handledare eller använd den **Tutorial** som finns under **Help**. Bläddra igenom menyerna för att få en allmän uppfattning om programmet.


2 Uppgifter

Det är ganska många uppgifter i denna laboration. Om ni inte hinner klart med allt på två timmar går det bra att fortsätta nästa vecka.

Uppgift 1. Skapa en katalog för denna laboration

I de följande övningarna kommer vi att skapa ett antal filer. För att samla dem på ett ställe och hålla isär dem från alla andra filer börjar vi med att skapa en ny katalog. Kalla den `Lab9` eller något liknande. (När senare uppgifter refererar till `Lab9` är det denna katalog som menas.)


Uppgift 2. Gör i ordning början på redovisningssidan

 I den gamla vanliga katalogen, `~hallgren/Intro/`, finns filen `labredovisning.html`. Kopiera den till katalogen `Lab9`. På några ställen står det `XX`. Byt ut dessa mot lämpliga uppgifter.

Kontrollera med Netscape att det ser bra ut.

Uppgift 3. En lathund över UNIX-kommandon

I laborationerna har vi nu stött på omkring 40 olika UNIX-kommandon. Det kanske är lite jobbigt att komma ihåg utantill hur alla fungerar. En liten lathund som sammanfattar kommandona kan vara till stor hjälp.

 Skapa en lathund över kommandona som använts. Den ska innehålla en enradsbeskrivning av varje kommando. Det finns en länk till en påbörjad lathund på kursens hemsida. Kopiera den till katalogen `Lab9` och utgå från den. Gör en länk från redovisningssidan till lathunden. Kontrollera att länken fungerar.

Det är tillåtet att samarbeta med en annan labbgrupp.

Uppgift 4. Låna en bild

Bilder från WWW-sidor kan man plocka ut och spara för egen användning. Detta får man dock bara göra om man vet att bilden är fri. Framför allt skall man akta sig för att ta bilder från företagssidor, inklusive bl a tidningssidor.

Man kan göra på två sätt:

1. Kopiera bilden och länka till den egna kopian. I Netscape gör man enklast så här: Placera markören på bilden. Tryck på musens högra knapp och välj **Save Image As...** i den meny som kommer fram. Spara bilden i lämplig katalog med lämpligt namn.
2. Låta bli att kopiera bilden och länka till originalet. Man kan få adressen till bilden genom att välja **Copy Image Location** menyn som fås med högra knappen.

☞ På www.iconbazaar.com finns det bilder som det är tillåtet att kopiera. Leta rätt på någon kul bild och stoppa in den i redovisningssidan. Skriv också en liten kommentar om var bilden kommer ifrån och vilka villkor som gäller för användandet av dessa bilder.

Uppgift 5. Färgtabellen

De flesta datorerna på E har ett system för färggrafik som gör att man inte kan ha mer än 256 olika färger samtidigt på skärmen. (Bildminnet lagrar bara en byte per pixel.) Palettens innehåll kan dock ändras. Även persondatorer som Mac och PC brukar ha detta system, men det brukar på dem numera också vara möjligt att ställa om till ett annat system som innebär att man kan ha många fler färger samtidigt.

Titta på färgtabellen med kommandot `xcmmap`. Förmodligen ser du en ganska slumpmässig blandning av olika färger. Sitter du däremot på en av datorerna med 24-bitarsgrafik visar `xcmmap` antagligen bara nyanser av en och samma färg.

☞ Gör en skärmbild av färgtabellen och ta med i redovisningsfilen.

Uppgift 6. Bildbehandling med xv

- Öppna filen `~hallgren/Intro/bild1.gif`. Spara den sedan i katalogen `Lab9`. Experimentera sedan med bildbehandlingsoperationerna under **Algorithms**. Du kan ångra med **Undo All**.
- Öppna filen `~hallgren/Intro/bakelse.jpg`. Som du ser är bilden för mörk och bakelsen ser inte så aptitlig ut. Det vanligaste problemet när bilder är för ljusa eller för mörka är att det s k gamma-värdet inte stämmer. Lägg in en gammakorrigering (omkring 1.4 är antagligen lagom), halvera bildens storlek och spara i katalogen `Lab9`.

☞ • Lägg till de båda bilderna i redovisningsfilen.

Kommandot `man xv` ger ett närmast innehållslöst manualblad, som dock leder dig till en mycket omfattande manual i PostScript (adressen är fel, skall fn vara `/usr/local/pd/inst/xv-3.10a/lib/xvdocs.ps`).

Man kan läsa om gammavärden på

<http://www.vtiscan.com/~rwb/gamma.html>

Uppgift 7. Ändra bakgrundsbild

Som du säkert har sett om du har surfat på webben, så kan man ha bakgrundsbilder på webbsidor. Detta gör man genom att stoppa in adressen till bilden i attributet `BACKGROUND` i dokumentets `BODY`-elementet, dvs man skriver

```
<BODY background="...">
```

där ... byts ut mot bildens adress. I *GNNPress* kan man lägga in bakgrundsbilder med menyalternativet **Format/Page Attributes...**

- ☞ Leta upp en bakgrundsbild på www.iconbazaar.com (eller på något annat ställe) och lägg in den i redovisningsdokumentet. Välj en *ljus* bild med ett *dämpat* mönster, så att texten fortfarande är lättläst.

Uppgift 8. Bilder med transparent bakgrund

Du har förmodligen noterat att i andras HTML-dokument finns det bilder liknande den mellersta nedan, medan du själv bara lyckats åstadkomma typen till vänster. Skillnaden är att bakgrunden lyser igenom allt som är vitt. GIF-formatet tillåter oss att lagra information om vilken färg som skall vara transparent. I den mellersta bilden är vitt transparent, medan i den högra grönt är det.



Programmet `giftrans` (som finns på det vanliga stället) kan användas för att göra transparent bakgrund. Det används så här:

<code>giftrans -l E.gif</code>	Visar färgtabellen i GIF-bilden <code>E.gif</code> .
<code>giftrans -t index E.gif >ETRANS.gif</code>	Gör en kopia av bilden med <code>index</code> (0-255) som transparent färg.
<code>giftrans</code>	Ger information om andra möjligheter.

- ☞ Gör en transparent kopia av en bild du ritat själv med `xpaint` eller av `~hallgren/Intro/E.gif`. Prova med *Netscape*. Lägg in både den ursprungliga bilden och den transparenta i redovisningsfilen.

Uppgift 9. Animerad GIF

Titta på sidan `file:/u1/ext/hallgren/Intro/animerad_gif.html` med *Netscape*. Den typ av bilder som visas kallas animerade GIF-bilder. Lagringsformatet GIF tillåter att flera bilder lagras i en fil.

Uppgift 10. Hämta program via Internet

Om man inte har ett program för lösa ett visst problem, t ex skapa animerade GIF-bilder, kan man i många fall hitta ett på Internet. När det gäller Mac/PC är programmen man hämtar ofta klara att köras. När det gäller UNIX-program är det av olika anledningar vanligt att man får källkoden till programmet. Man måste då kompilera det själv. Ofta är programmen skrivna i C, eftersom en C-kompilator ingår som standard i de flesta UNIX-system.

Det finns ett program som heter *WhirlGIF* som tillverkar animerade GIF-bilder. Det finns på webbsidan

```
http://www.msg.net/utility/whirlgif/
```

Hämta hem det (filen heter `whirlgif.tar.gz`) och spara det i katalogen `Lab9`. Filnamnsslutet `.tar.gz` antyder att det är ett tar-arkiv som komprimerats med `gzip`. Programmet `tar` (tape archive) utformades från börjar för att göra backup av filer (vanligen hela kataloger med filer och allt) på band, men används även när man vill packa ihop många filer till ett paket. (Se man `tar` för uttömmande information.)

Packa upp filen med kommandona

```
gunzip whirlgif.tar.gz  
tar xvf whirlgif.tar
```

En ny katalog som heter `whirlgif` skapas. Flytta dig in i den. Se efter vilka filer som finns där. Du kan kompilera källkoden genom att bara skriva

```
make
```

Då skapas ett körbart program `whirlgif` med vilket animerade GIF-bilder kan bildas. Titta hastigt på `whirlgif.doc` som beskriver hur man går tillväga.

Uppgift 11. Tillverka animerad GIF-sekvens

Stanna kvar i katalogen ovan. Skapa 2-3 GIF-bilder med `xpaint` (alternativt med *Maple* eller *MATLAB*) som skall bilda underlag för animeringen. Kalla dem t ex *G1.gif*, *G2.gif* och *G3.gif*. Skriv sedan (omdirigering!)

```
whirlgif G*.gif > ANIMG.gif
```

så bildas den animerade filen. (`G*.gif` är ett mönster som ersätts med passande filnamn av kommandotolken innan kommandot utförs. Här får vi alla filer vars namn börjar på *G* och slutar på `.gif`.)



Lägg in den animerade bilden i redovisningsdokumentet och kontrollera att animeringen fungerar i *Netscape*.

Uppgift 12. Klickbara kartor med GNNPress

Gör ett nytt HTML-dokument med *GNNPress* genom att välja **File/New Page** i menyn. Fyll i filnamn och titel. Välj därefter **Image** och låt den utgöras av `http://www.chalmers.se/GIF/CTH-karta-www.GIF`. Vi skall nu göra så att när man klickar på kartan i närheten av E så skall man komma till E-s hemsida

<http://www.etek.chalmers.se> och när man klickar nära Matematiskt centrum skall man komma till <http://www.cs.chalmers.se>. Markera kartan genom att dubbelklicka. Välj **Element/Image map...** från menyn. Det kommer upp ett nytt fönster. Välj något av de tre verktygen rektangel, cirkel och polygon. Drag kring E. Fyll i adressen i rutan **Location**. Drag på samma sätt kring Matematiskt Centrum och fyll i adressen. Spara med **SAVE**. Avmarkera bilden. För markören över bilden. Klicka nära något av de två ställena. Du märker förmodligen nu om inte tidigare att *GNNPress* inte är sämre än Netscape på vanlig webbläsning.

☞ Spara ditt dokument i katalogen Lab9. Gör en länk till det från redovisningsdokumentet. Prova att länken och den klickbara kartan fungerar i *Netscape*.

Uppgift 13. Interaktiva webbsidor och JavaScript

Animerade GIF-bilder kan göra webbsidor mer levande, men betraktaren kan ändå inte göra så mycket mer än att titta på sidan och följa länkar.

Man kan skapa webbsidor med lite mer interaktiva inslag genom att utöka HTML-dokumentet med ett avsnitt med kod i JavaScript. Här visas ett enkelt exempel, där användaren matar in ett uttryck, trycker på knappen **Räkna ut** och får resultatet.



JavaScript-koden, som finns i filen `~hallgren/Intro/javascript.html`, är i detta exempel uppdelad i två delar. Den första delen är en hjälpfunktion som ligger i dokumentets HEAD-del:

```
<SCRIPT LANGUAGE="JavaScript">
function compute(form) {
    if (confirm("Är det verkligen detta som du vill ha
beräknat?"))
        form.result.value = eval(form.expr.value)
    else
        alert("Det var väl kul. Försök gärna igen!")
}
</SCRIPT>
```

Den andra delen finns i BODY-delen och skapar de synliga delarna och använder hjälpfunktionen:

```
<FORM>
Skriv ett uttryck:
<INPUT TYPE="text" NAME="expr" SIZE=15 >
<INPUT TYPE="button" VALUE="Räkna ut"
ONCLICK="compute(this.form)" >
<BR>
Resultat:
<INPUT TYPE="text" NAME="result" SIZE=15 >
</FORM>
```

- ☞ Kopiera in detta JavaScript-exempel i ditt redovisningsdokument. Detta görs lämpligen med `nedit`. Om du har redigerat din webbsida med *GNNPress* innan är det bäst att spara och stänga dokumentet i *GNNPress* först.

Uppgift 14. Applets

Applets är ett annat sätt att göra interaktiva inslag i webbsidor som först dök upp i Suns webbläsare HotJava. Applets är småprogram som vanligen skrivs i språket Java, som är så populärt just nu. Gå till

```
http://www.javasoft.com/applets/
```

och titta t ex på appletterna *Pythagoras* och *Nuclear Plant* under rubriken *Applet Archive*.

- ☞ Lägg också märke till att det finns instruktioner om hur man lägger till en applet till en webbsida. Leta upp en applet som du tycker om och kopiera den till redovisningssidan!

Uppgift 15. Tillverka en logotyp

Det finns ett bildbehandlingsprogram för UNIX som heter GIMP (se www.gimp.org). Det är gratis och liknar programmet PhotoShop (som inte är gratis). GIMP är programmerbart och folk har gjort diverse små program för att tillverka tju-siga logotyper, knappar, animeringar och annat kul. GIMP finns inte på E, men det finns en webbsida, www.cooltext.com, där man kan tillverka egna logotyper genom att fjärrstyra GIMP.

- ☞ Tillverka en logotyp och lägg in den överst på redovisningssidan.

Uppgift 16. VRML

WWW har gett upphov till en mängd idéer. En sådan är att man skall kunna hämta 3D-modeller från webbservern (precis som bl a vanliga HTML-sidor) och sedan interaktivt kunna vandra runt i dessa världar. Det finns en framväxande standard för sådana modeller i form av ett språk kallat VRML (Virtual Reality Modelling Language). Starta programmet `vrweb` och öppna dokumentet `~hallgren/Intro/BANAN.wrl`. Utforska menyerna och experimentera. Öppna sedan det större dokumentet `~hallgren/Intro/GRAZ.wrl` och lek litet. Titta gärna även på det första dokumentet med t ex *NEdit* eller lista det bara i fönstret med `cat`.

Anm. VRML-dokument kan också öppnas från *Netscape* om man har rätt inställningar.

Uppgift 17. Multimedia: xanim

X-programmet `xanim` kan visa animerade GIF-bilder och en mängd andra format för rörliga bilder, bl a MPEG, AVI (från Microsoft) och QuickTime (från Apple), som även kan innehålla ljud. Prova denna aktuella trailer (som jag hittade via The Internet Movie Database på www.imdb.com):

```
xanim +Sr ~hallgren/Intro/trumanhi.mov
```

Du kan stoppa/starta bildväxlingarna med mellersta musknappen. Vänstra/högra stegar en bild bakåt/framåt. Flaggan +SR gör att man kan ändra storlek på fönstret medan filmen spelas. Programmet kan stoppas med q i filmfönstret.

3 Avslutande kommentarer

Du som är mycket intresserad av fler detaljer kring det som tas upp här hänvisas till det rikliga bokutbudet eller kanske hellre till det ofta mycket aktuella material som finns på nätet.

3.1 Information på webben

Redan i anslutning till de olika uppgifterna har det dykt upp ett antal webbadresser. Här följer ytterligare några.

På www.htmlhelp.com kan man hitta diverse nyttig information för webbsidemakare, bland annat kortfattade och överskådliga beskrivningar av HTML.

När man bläddrar i svenska Yahoo (www.yahoo.se) hittar man en del HTML-kurser, nybörjarguider och annat på svenska. Två exempel är:

<http://www.april.se/htmlkurs/>

<http://www.skolverket.se/skolnet/htmlkurs/>

Om man söker i Yahoo eller andra webbkataloger hittar man ganska många webbsidor med samlingar av symboler, linjer, prickar, animerade GIF-bilder och annat som man kan utsmycka sina webbsidor med. Vi har redan sett www.iconbazaar.com. Ett annat exempel är www.mediabuilder.com.

Genom att söka efter JavaScript i Yahoo hittar man lätt många JavaScript-exempel.

När man gör webbsidor testas man förstås att de ser bra ut i den egna webbläsaren, men olika webbläsare hanterar felaktig HTML-kod på olika bra, så det kan vara bra att kontrollera att HTML-koden följer standarden. Det kan man göra på webbsidan validator.w3.org. Det finns också ett program som heter *weblint* som kontrollerar HTML-filer, men det verkar inte vara installerat på E.

På www.useit.com skriver Jakob Nielsen om hur man gör bra webbsidor, vad som är bra och dåligt på webben, m m. I hans "AlertBox" hittar man t ex artiklarna *Why Yahoo is good*, *Top ten mistakes of web design*, *Why frames suck*, m m. Exempel på en svensk webbplats där webbdesignfrågor diskuteras är

<http://www.idg.se/klubben/webstudio/>

Laborationer i kursmomentet Datoranvändning E1

<http://www.etek.chalmers.se/~hallgren/Eda/>

Laboration nr 10: Programmering i Maple

1996, 1997 Magnus Bondesson
98-11-23 Thomas Hallgren

1 Introduktion

Syftet denna gång är att ge dig en viss inblick i vad programmering är och hur den går till samt vad vi kan uppnå med sådan. Vi arbetar i Maple men förhoppningen är att du skall få en allmän förståelse. Laborationen går utöver vad som nämns i Gula Boken. Jag försöker dock här att ta med alla erforderliga huvuddrag.

Det ingår också övningar på några användbara UNIX-kommandon, denna gång `which`, `jobs`, `kill` och `ps`.

Detta är den sista laborationen på kursen. Den ska helst utföras i läsvecka 5 (23-27 Nov), men det kommer i begränsad omfattning att finnas handledare tillgängliga även i läsvecka 6. Sista handledda labbpasset (och sista möjligheten att bli godkänd) är på torsdagen den 3 december.

1.1 Förberedelser

Innan du kommer till laborationen är det lämpligt att ha tittat på några sidor i Gula Boken, denna gång avsnitt 9.8 och 9.4.8. Läs också åtminstone avsnitt 2 i detta laborations-PM och fundera igenom några av de inledande Maple-uppgifterna hemma.

1.2 Redovisning

Skriv svaren på alla Maple-uppgifterna i ett och samma Maple-dokument. Spara ofta, så att inte allt går förlorat om Maple skulle krascha. UNIX-uppgifterna redovisas genom svar på frågorna i detta laborations-PM.

2 Om programmering

Programmering handlar om att i vid mening automatisera ett förfarande, dvs i stället för att manuellt genomföra ett antal åtgärder, så skriver vi ett **program** som genomför dem. I alla problemlösningssverktyg finns programmeringsmöjligheter, bl a av det skälet att användaren skall kunna lösa nya problem och inte bara de som verktyget har mer eller färdiga lösningar till.

Det finns en del begrepp som är centrala för all programmering:

- Variabel

- Styrkonstruktionerna räknarsnurra och villkorssnurra
- Styrkonstruktionen villkorligt utförande
- Funktioner/procedurer
- Parametrar
- Rekursion
- Konstruktioner som gör att man klarar sig utan t ex snurror ibland. I Maple finns t ex *seq*, *map*, *add*, *sum*, *mul*, *product* m fl.

Hur dessa ser ut i just Maple framgår av Gula Boken, men låt oss ändå samla huvuddragen här. För detaljer hänvisas till boken.

Räknar-snurrans form:

```
for variabel from start by steg to slut do
    satser ;
od;
```

Semikolonet efter *satser* behövs egentligen inte här eller i det följande och jag sätter inte alltid ut det i exemplen.

Exempel

```
# Beräknar summan 1+2+3+ ... + 10
s:=0;
for i from 1 to 10 do
    s:=s+i;
od;
```

Inmatning av sådana här konstruktioner gör man i praktiken på något av de tre sätten (av de redigeringskäl som vi redovisat tidigare under kursen):

- Om den är mycket kort: På en rad.
- Om den är litet längre: Rad för rad med SHIFT/RETURN-tangenterna mellan raderna. Den sista raden avslutas med retur-tangenten varvid konstruktionen utförs.
- Om den är längre: Mata in den i något redigeringsprogram, t ex *NEdit*. Spara under lämpligt namn. Skriv i Maple: `read `filnamnet` ;` och konstruktionen utförs.

Från Maples synpunkt kan kod skrivas in hur som helst, t ex med allt på en rad eller ett ord per rad. Men för att koden skall bli läsvänlig mänskligt sett, bör man **strukturer** den så som vi visar i ovanstående och följande modeller.

Villkors-snurrans form:

```
while villkor do
    satser ;
od;
```

Det finns också en kombinerad version som inleds med *for ... to slut* while *villkor* do.

Exempel

```
# Beräknar summan 1+2+3+ ... + 10
s:=0;i:=1;
while i<=10 do
    s:=s+i;i:=i+1;
od;
```

Villkorskonstruktionen kan ta några olika former

```
if villkor then          if villkor then          if villkor1 then
    satser ;              satser ;              satser ;
fi;                      else
                          satser ;
                          fi;
                          elif villkor2 then
                          satser ;
                          else
                          satser ;
                          fi;
```

Exempel

```
# Bestämmer det största av två tal x och y, max är ett upptaget namn
if x>y then
    maxi:=x;
else
    maxi:=y;
fi;
```

En **funktionsdefinition** kan ges på ett par olika former. Den första har vi tidigare mött och kan användas när beräkningen inte behöver några hjälpvariabler.

```
funktionsnamn:=(parametrar)->uttryck i parametrarna;
```

Om det bara finns en parameter, behövs inga parenteser. Behövs hjälpvariabler (dessa införs i *local*-delen) får man ta till den allmännare formen

```
funktionsnamn:=proc(parametrar)
    local lokala_namn;
    satser ;
end;
```


Definitionen "ekas", vilket kan hindras om man ersätter det avslutande semi-kolonet med ett kolon. I båda fallen sker anrop med

```
funktionsnamn(aktuella parametrar);
```

Exempel

```
> f:=x->x^5;
      f := x -> x^5
> f(2);
      32
> f:=proc(x)
      x^5;
      end;
      f := proc (x) x^5 end
> f(2);
      32
> f:=(x,n)->x^n;
      f :=(x, n) -> x^n
> f(2,4);
      16
> f:=x->if x<0 then -x else x fi;
f := proc (x) options operator, arrow; if x < 0 then -x else x fi end
> f(-3);f(2);
      3
      2
> f:=(x,y)->x^2+y^2;
      f := (x, y) -> x^2+y^2
> f(1,2);
      5
> summa:=proc(n)
      local s,i;
      s:=0;
      for i from 1 to n do
          s:=s+i;
      od;
      end;
summa := proc (n) local s, i; s := 0; for i to n while true do s :=
s+i od end
> summa(10);
      55
> summa:=n->if n=0 then 0 else n+summa(n-1) fi;
summa := proc (n) options operator, arrow; if n = 0 then 0 else
n+summa(n-1) fi end
> summa(10);
      55
```

Utskrifter i styrkonstruktioner och funktioner kan vålla en del problem, se sid 186 i Gula Boken. I korthet: Om du tycker att du får för många utskrifter, ändra systemvariabeln *printlevel* till 0 eller -1 före konstruktionen (återställ till standardvärdet 1 efteråt). Om du saknar en utskrift av något, framtvinga den genom att skriva

`print(de saker du vill ha utskrivna)`. Känn dig inte tvingad att ta bort överflödiga utskrifter i uppgifterna.

3 Några erinringar angående Maple

Vid det här laget har du förmodligen glömt en del. Observera att tilldelningstecket i Maple är ":@" och att varje kommando skall avslutas med ";" (eller ":" för undertryckande av utskrift). Vid redigering flyttar man sig upp i den tidigare textmassan med musen (markören fixeras med **vänstra mustangenten** inte med någon av de andra två, som har med klistring att göra) eller piltangenter och vid utförandet av ett tidigare kommando (eventuellt redigerat) ersätts det tidigare svaret. Med musen eller piltangent kan man återvända till slutet. Undvik att trycka på returtangenten om något i textmassan är markerat; det markerade tas nämligen som ett kommando. Observera också att liksom i UNIX betyder i Maple stor och liten bokstav olika sak (oftast).

4 Uppgifter

Laborationstiden räcker säkert inte till för att göra allt. Uppgifterna bör göras i angiven ordning och *-märkta uppgifter i första hand. Övriga uppgifter är frivilliga.

4.1 Först några små UNIX-uppgifter

*Uppgift 1. Var finns alla kommandon?

Kommandon i UNIX är program, som finns i filer. Med kommandot `which` kan man ta reda på var ett kommando finns. Kommandot `which ls` ger svaret `/usr/bin/ls`, t ex.

- Var finns kommandot `date`? _____
- Var finns kommandot `v`? _____
- Var finns `netscape`? _____

I de tidigare laborationerna har vi då och då använt program från katalogen `~hallgren/Intro/` och det har varit lite jobbigare än att använda vanliga kommandon, som t ex `ls`. Hur kommer det sig att det räcker att skriva `ls` för att kommandotolken ska hitta `/usr/bin/ls` när det inte räcker att skriva `giftrans` får att kommandotolken ska hitta `~hallgren/Intro/giftrans`? Svaret är att kommandotolken bara letar i de kataloger som anges av variabeln `$path`.

- Hur många kataloger letar kommandotolken i? _____

Man kan lätt själv ändra vilka kataloger kommandotolken ska leta i. Antag t ex att vi vill lägga till `~hallgren/Intro/` i listan. Vi kan då skriva

```
set path = ( $path ~hallgren/Intro/ )
```

vilket betyder att det nya värdet på variabeln blir det gamla utökat med `~hallgren/Intro/`.

Anm. Om det finns två kommandon med samma namn får man det som finns i den katalog som står först i listan.

Anm. I hemkatalogen finns en dold kommandofil som heter `.cshrc`. Kommandona i den utförs varje gång en ny kommandotolk startas (t ex när man startar ett terminalfönster). Man kan stoppa in egna kommandon här, t ex för att lägga till egna kommandokataloger till `$path`-variablen.

*Uppgift 2. Att stoppa skenande program

Detta är till viss del repetition av Uppgift 12 och Uppgift 13 i Laboration nr 2. Se även dessa uppgifter.

Ibland händer det att program man startat inte fungerar så bra. Ibland går det kanske t om så galeat att det normala sättet att avsluta programmet inte fungerar. Man kan då använda kommandot `kill` för att skicka en stoppsignal till programmet. Kommandot används så här:

<code>kill %jobbnummer</code>	Kommandot <code>jobs</code> ger en lista över de program som startats i <i>samma</i> terminalfönster. Denna lista innehåller <i>jobbnummer</i> , som man kan använda om man vill avbryta program.
<code>kill processnummer</code>	Med kommandot <code>ps</code> kan man få en lista över alla program som körs på datorn (egna eller egna+andras). Listan innehåller <i>processnummer</i> .
<code>kill -9 ...</code>	Avslutar ett program med våld. Programmet får ingen chans att städa upp efter sig.

Se även manualbladen för `kill` och `ps`.

Om man startar ett program som använder för mycket minne blir ofta hela datorn väldigt seg och man kanske inte kan göra någonting, inte ens starta ett terminalfönster för att kunna göra `kill` på programmet. Det kan också hända att ett program låser fönstersystemet. Man kan då använda `rlogin` eller `slogin` (se Laboration nr 8) från ett terminalfönster på en annan dator och köra `kill` på det.

- Kör programmet `~/hallgren/Intro/mystiskt`. Det kommer att starta en process som du ska avbryta med `kill`. Du kommer att behöva använda `ps` för att ta reda på processnumret. Mer instruktioner skrivs ut då programmet körs.

När du har avbrutit programmet får du ett kvittensmeddelande. Skriv ner det här:

4.2 Maple-uppgifter

För att få roligare uppgifter kommer vi successivt att introducera ytterligare detaljfakta om Maple. Meningen är därmed inte att du skall känna dig tvingad att lära dig dessa utantill och de är inte huvudsyftet med uppgifterna, varför jag försöker ge så mycket hjälp som möjligt. Fråga även handledaren vid behov.

Lösningar dyker upp på kursens hemsida så småningom.

***Uppgift 3. En funktion - repetition**

Skriv en Maple-funktion motsvarande $f(x)=x^4-1$. Kalla den t ex FUNK1. Bilda även derivatan `FUNK1DER:=D(FUNK1)`;

***Uppgift 4. En räknarsnurra**

Skriv satser som beräknar och skriver ut $[x, f(x), f'(x)]$ där f är funktionen i förra uppgiften för $x=0, 0.1, 0.2, 0.3, \dots, 0.9, 1.0$.

***Uppgift 5. Collatz problem med WHILE och IF**

Låt N vara ett positivt heltal större än 1. Om det är jämnt (kan avgöras med att man testar om $N \bmod 2 = 0$), halvera det, i annat fall multiplicera det med 3 och lägg till ett. Om det nya talet är 1, är vi nöjda. Upprepa annars förfarandet med det nya. Det har praktiskt visat sig att man alltid når 1, men ingen har lyckats bevisa det. Exempel: $N=5$ ger följden: 5,16,8,4,2,1. Skriv satser som givet N skriver ut följden i form av en sekvens eller lista.

Uppgift 6. Primtalstvillingar med WHILE och IF

Två heltal N och $N+2$ kallas primtalstvillingar om båda är primtal. Skriv satser som bestämmer det första tvillingparet som är större än 700. Gå igenom de tänkbara kandidaterna och bryt när det önskade paret hittats. Påminnes om att funktionen *isprime* avgör om ett tal är primtal eller ej.

Uppgift 7. Primtalstvillingar, forts. med FOR och IF

Skriv satser som på formen `[17,19]` skriver ut samtliga primtalstvillingar mindre än 2000 och även räknar antalet sådana.

***Uppgift 8. En funktion som ger sista siffran i ett positivt heltal**

Skriv en funktion *Sista(N)* som ger sista siffran i heltalet N . Sista siffran fås som heltalsresten vid division med 10, vilket Maple ger med $N \bmod 10$.
Exempel: `Sista (184)` ger 4.

***Uppgift 9. En funktion som ger första elementet i en lista**

Någon sådan behöver vi inte eftersom första elementet i en sekvens eller lista L kan fås med `L[1]`, men gör ändå en sådan funktion `First(L)`. Ex: `First([8,1])` ger 8.

Uppgift 10. En funktion som ger de två första elementen i en lista

Skriv en funktion *FirstPair(L)*, vars värde är listan bestående av de två första elementen i listan L .

***Uppgift 11. En funktion som kastar om elementen i en tvåelementlista**

Skriv en funktion $Change(L)$, vars värde är en lista med elementen $L[1]$ och $L[2]$ omkastade. T ex skall $Change([7, 9])$ ge $[9, 7]$.

***Uppgift 12. En funktion som summerar elementen i en lista av tal**

Skriv en funktion $Summa(L)$, vars värde är summan av alla elementen (som förutsättes vara tal) i L . T ex $Summa([3, 1, 9])$ skall ge 13.

***Uppgift 13. En egen funktion**

Gör en egen funktion som beräknar N-fakultet, dvs fungerar som den inbyggda *factorial*. Funktionen skall alltså beräkna $1*2*3*...*N$.

***Uppgift 14. En funktion som letar i en lista**

Skriv en funktion $Leta(L, value)$, som letar igenom listan L efter värdet $value$ och returnerar motsvarande platsnummer (0 om värdet inte finns med). T ex $Leta([7, 5, 13, 8], 13)$ ger 3.

Uppgift 15. En funktion behöver inte beräkna ett egentligt värde

Skriv en funktion $Rita(f)$, som ritar upp funktionen $f(x)$ på intervallet $[-1, 1]$. T ex skall $Rita(x \rightarrow x^2)$ rita en del av en parabel.

***Uppgift 16. Automatisera Newtons metod**

Newtons metod för att bestämma ett nollställe till en funktion $f(x)$ består väsentligen i Maple i man upprepar

$$x := x - f(x) / D(f)(x);$$

ett antal gånger. Först måste vi naturligtvis ha definierat funktionen och gett x ett startvärde. Automatisera beräkningarna med en snurra. Skriv för varje iterationsvärde ut dels det nya x -värdet, dels skillnaden mot det gamla. **Tips:** `printlevel:=0` före snurran och `print([x, x-xold])` ger precis de önskade utskrifterna.

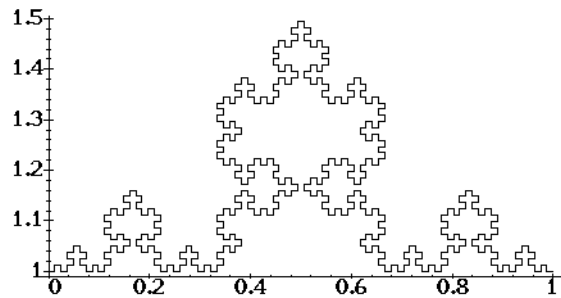
Uppgift 17. En funktion som bestämmer ett nollställe till en godtycklig funktion

Skriv nu en Maple-funktion $Newton(f, x0)$ som beräknar ett nollställe till $f(x)$ i närheten av $x0$ (eventuellt komplext; metoden fungerar utan ändringar precis lika bra då; för att komplexa nollställena måste man starta med komplext $x0$). Prova den på funktionen $f(x)=x^4-1$ som ju har de fyra nollställena $1, -1, i, -i$.

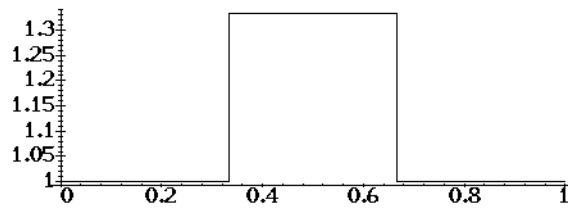
Låt beräkningen avbrytas då skillnaden mellan de två senaste värdena är liten (t ex $abs(x-xold) < 0.00001$), men gör maximalt 20 iterationer (flera villkor kan kombineras med `and`).

Uppgift 18. Välkommen till fraktalernas värld

Kurvan här intill är ett exempel på en så kallad fraktalkurva, ett begrepp som myntades av den fransk-amerikanske matematikern Mandelbrot i 1975. Fraktal betyder bruten och det är ju synbart en egenskap som kurvan har. Kurvan börjar i punkten (0,1) och slutar i (1,1). Det är inte ovanligt att man i programmeringskurser med grafikinslag tar upp ritandet av sådana. Dels därför att det kan vara skojigt, men framför allt för att rekursion är en behändig teknik i sammanhanget. Det går bra även i Maple, men man får tänka litet annorlunda än i ett traditionellt programspråk och tidsåtgången blir betydligt större.



Låt oss först förklara idén. Den ovan ritade kurvan är 4:e generationen i en familj vars 0:te generation utgörs av ett rakt horisontellt streck från (0,1) till (1,1) och vars 1:sta generation är kurvan (kallad generatoren eftersom den bestämmer de följande generationernas utseende).



Nästa generation, den andra, får vi genom att ersätta de fem streck som bygger upp denna kurva med en förminskad version av generatoren, precis som vi fick den första generationen genom att ersätta det ursprungliga strecket med generatoren. Låt P_0 och P_1 vara start- och slutpunkt på ett generatorsegment och låt P_2 , P_3 , P_4 och P_5 vara de mellanliggande brytpunkterna (markerade för hand i figuren ovan). Om vi nu vill rita ett visst generationsnummer N , ritar vi helt enkelt närmast lägre nummer mellan P_0 och P_2 , mellan P_2 och P_3 , mellan P_3 och P_4 , mellan P_4 och P_5 och till sist mellan P_5 och P_1 . Detta är en rekursiv process, som lätt låter sig formuleras. En mindre svårighet är att beräkna koordinaterna för punkterna, så därför ger jag dem här (vi representerar punkter med en tvåelementslista $P=[x,y]$, dvs $P[1]$ är x-koordinaten):

```
dx:=(P1[1] - P0[1])/3: dy:=(P1[2] - P0[2])/3:
P2:=[P0[1]+dx, P0[2]+dy]: P3:=[P2[1]+dy, P2[2]+dx]:
P5:=[P1[1]-dx, P1[2]-dy]: P4:=[P5[1]+dy, P5[2]+dx]:
```

Skriv nu en rekursiv funktion $fract(N,P_0,P_1)$ som genererar en sekvens S av samtliga brytningspunkter för kurvan i den N :te generationen, $N \geq 0$. Kurvan för $N=4$ kan sedan ritas med

```
plot([ fract( 4,[0,1],[1,1] )], scaling=constrained);
```

Jag påminner om att två sekvenser S_1 och S_2 kan sättas ihop till en enda med S_1,S_2 .

Uppgift 19. Beräkna π med många siffror

En nyligen (under 1980-talet) upptäckt snabb metod för beräkning av talet π med massor av decimaler lyder som följer

1. Sätt $y_0 = \sqrt{2} - 1$ och $\alpha_0 = 6 - 4\sqrt{2}$

2. Upprepa för $n=0,1,2,3,\dots$

2.1. $y_{n+1} = \frac{1-b}{1+b}$, där $b = (1-y_n^4)^{1/4}$

2.2. $\alpha_{n+1} = (1+y_{n+1})^4 \alpha_n - 2^{2n+3} y_{n+1} (1+y_{n+1}+y_{n+1}^2)$

Då närmar sig $p=1/\alpha_n$ talet π mycket snabbt när n växer; antalet riktiga siffror mer än fyrdubblas per upprepningssteg. Verifiera detta praktiskt med Maples hjälp enligt:

1. Sätt antalet siffror som Maple arbetar med till t ex 500 med `Digits:=500;`.
2. Vi behöver tre variabler y , a och p för y_n , α_n och p ovan. Ge y och a startvärden ($n=0$).
3. Beräkna i en räknarsnurra med 5 varv $y = y_n$ och $a = \alpha_n$ samt p , för $n=1,2,3,4,5$. Beräkna med `evalf(p-Pi)` felet och skriv i varje varv ut p och felet. Notera hur felets storleksordning avtar.
4. Återställ antalet arbets-siffror med `Digits:=10;`

5 Avslutande kommentarer

Detta är inte en programmeringskurs, så meningen med denna laboration är som sagt bara att ge en liten inblick i vad programmering kan innebära. E:s programmeringskurs ligger i årskurs 2.

5.1 Information på webben

Hemsidan för kursen Programmeringsteknik E finns på adressen

<http://www.cs.chalmers.se/Cs/Grundutb/Kurser/ept/>

Andra kurser som ges av Institutionen för Datavetenskap finns på

<http://www.cs.chalmers.se/Cs/Grundutb/Kurser/>

Den historiska utvecklingen av datorer och datavetenskap sammanfattas på

<http://math.uwaterloo.ca/~shallit/Courses/134/history.html>

För den som är intresserad av hur program ser ut i olika programspråk rekommenderas sidan

<http://www.ionet.net/~timtroyr/funhouse/beer.html>

där ett och samma program finns översatt till över 200 olika programspråk.