

Datorer och datoranvändning

Föreläsning 2 — L^AT_EX

Mattias Nordahl

`mattias.nordahl@cs.lth.se`

Föreläsning 2 — L^AT_EX

Förberedelse inför laboration 2.

Ordbehandling

L^AT_EX

Mall för rapport

Dokumentstruktur: dokumentklasser, omgivningar, text, stycken, listor, tabeller, ...

Programlistor

Matematiska formler

Bilder

De flesta moderna ordbehandlare, till exempel Microsoft Word, fungerar enligt WYSIWYG-principen:

What You See Is What You Get

Det innebär att det man ser på skärmen ser likadant ut som det som kommer att skrivas på papperet: teckensnitt, storlekar, avstånd, ... Det innebär också att det inte blir bättre än vad det ser ut på skärmen (What You See Is *All* You Get).

I de flesta ordbehandlare finns det formatmallar där man till exempel kan bestämma att alla rubriker på en viss nivå ska ha ett visst utseende. Om man vill ändra utseendet på alla rubriker så räcker det att ändra i mallen.

Det brukar också finnas möjlighet till automatisk numrering av rubriker, automatisk generering av innehållsförteckning och sakregister och liknande.

När man skriver matematisk text använder man ofta en ekvationseditor för att skriva de matematiska symbolerna. Ekvationseditorer är inte enkla att använda, och slutresultatet brukar inte bli bra.

Med L^AT_EX arbetar man på ett helt annat sätt: man skriver texten i en vanlig textfil och lägger in *kommandon* ("taggar") i texten som visar hur texten ska formateras. Textfilen kan bli något svårläst, åtminstone innan man är van, men resultatet blir garanterat snyggt.

Enkelt exempel:

Pythagoras sats ser ut så
här: $a^2 + b^2 = c^2$.

Pythagoras sats ser ut så här: $a^2 + b^2 = c^2$.

\$-tecknen anger att en matematisk formel börjar och slutar. L^AT_EX vet då att variablerna a , b och c ska skrivas kursiva, hur stora exponenterna ska vara och var de ska placeras, och hur mycket mellanrum det ska vara mellan termerna.

Ett större exempel

If f is continuous on the closed interval $a \leq x \leq b$ and differentiable on the open interval $a < x < b$, then there exists a point

ξ , $a < \xi < b$ such that

$$f(b) - f(a) = f'(\xi)(b - a).$$

If f is continuous on the closed interval $a \leq x \leq b$ and differentiable on the open interval $a < x < b$, then there exists a point ξ , $a < \xi < b$ such that

$$f(b) - f(a) = f'(\xi)(b - a).$$

Donald E. Knuth skrev 1977–1982 typsättningsprogrammet T_EX¹ eftersom han inte var nöjd med de möjligheter till typsättning som fanns då. T_EX är ett "lågnivåspråk". Leslie Lamport byggde på T_EX med ett makropaket som gör det möjligt för författaren av ett dokument att koncentrera sig på den logiska strukturen hos dokumentet och på själva texten i stället för på lågnivåtypsättningen. Resultatet blev L^AT_EX². En föregångare till L^AT_EX, *t_roff*, används fortfarande ibland, till exempel till Unix man-sidor.

¹T_EX skrivs TeX i skrivmaskinsskrift och uttalas "tech".

²L^AT_EX skrivs LaTeX i skrivmaskinsskrift och uttalas "lah-tekh" eller lay-tekh".

När man använder L^AT_EX utgår man från en fil med text och kommandon. Filen ska ha tillägget `.tex`, till exempel `rapport.tex`. Sedan "översätter" man filen till pdf-format med programmet `pdflatex` och tittar på resultatet med en pdf-läsare, till exempel `evince`. Detta kan man naturligtvis göra genom att skriva kommandona för hand (`gedit rapport.tex`, `pdflatex rapport.tex`, `evince rapport.pdf`), men det är enklare att använda ett specialprogram. På studentdatorerna finns programmet `texmaker`. På Mac-datorer använder man t.ex. `TeXShop` eller `TeXstudio` som även finns för Windows.

I stället för att generera pdf-filer med `pdflatex` kan man generera dvi-filer ("device independent") med programmet `latex` som man kan titta på med en "dvi-läsare" och sedan översätta till Postscript eller pdf. Numera använder de flesta `pdflatex`.


```
\documentclass[a4paper]{article}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[swedish]{babel}
\usepackage{fancyvrb}
\fvset{tabsize=4}
\fvset{fontsize=\small}
\title{Programmeringsteknik\
  Inlämningsuppgift 1}
\author{Xerxes Yngvesson\
  dat14xyn@student.lu.se}
\date{2014--10--17}

\begin{document}
\maketitle

Här skriver man texten i
rapporten.

\end{document}
```

Programmeringsteknik Inlämningsuppgift 1

Xerxes Yngvesson
dat14xyn@student.lu.se
2014-10-17

Här skriver man texten i rap-
porten.

Dokumentklasser och omgivningar

`{article}` är en *dokumentklass* (den man oftast använder). Andra dokumentklasser är `{report}`, `{book}`, `{letter}` och `{beamer}` (beamer används för overheadbilder). En dokumentklass påverkar utseendet på hela dokumentet.

`\begin{document}` definierar starten på en *omgivning*, `\end{document}` slutet på omgivningen. En omgivning påverkar utseendet på den del av dokumentet som ingår i omgivningen. Vi kommer att se exempel på andra omgivningar senare.

Radslut och antal mellanslag mellan ord har ingen betydelse, \LaTeX formaterar så att det blir snyggt. En eller flera blanka rader ger ett nytt stycke. Exempel:

Det här
är en text som jag
har skrivit. Det är
en lång text med flera
rader.

Här börjar det ett
nytt stycke i texten.

Det här är en text som jag har skrivit. Det är en lång text med flera rader.

Här börjar det ett nytt stycke i texten.

\LaTeX numrerar rubriker automatiskt. Man anger en rubrik med `\section` eller `\subsection`.

```
\section{Inledning}
```

```
\section{Utförande}
```

```
\subsection{Del 1}
```

```
\subsection{Del 2}
```

```
\section{Slutsatser}
```

1 Inledning

2 Utförande

2.1 Del 1

2.2 Del 2

3 Slutsatser

Ändra textens utseende

Det finns många kommandon för att ändra utseende på texten. Två sådana kommandon är `\emph` för att betona text och `\texttt` för att skriva med skrivmaskinstypssett. Exempel:

Här skriver jag något
`\emph{viktigt}`. Och
i Java har vi använt
klassen `\texttt{Square}`.

Här skriver jag något *viktigt*. Och i
Java har vi använt klassen Square.

Det finns också kommandon för fetstil, lutande text, osv, och för att ändra storlek på texten. Använd sparsamt!

Med tecknet % inleder man en kommentar som sträcker sig till slutet av raden.

En del tecken används för kommandon och måste skrivas på speciellt sätt:

```
\$ \% \_ \# \& \{ \} \textbackslashash
```

Det finns streck, mellanrum och punkter av olika slag:

```
DoD-kursen pågår under vecka  
1--3 av läsperiod ht1. Tyvärr  
är den inte längre \ldots
```

```
\quad Telefon: 046--222~80~38.  
Dagens datum: \today.
```

```
DoD-kursen pågår under vecka 1-3  
av läsperiod ht1. Tyvärr är den inte  
längre ...
```

```
Telefon: 046-222 80 38. Dagens  
datum: 6 september 2022.
```

Fotnoter är lätta att skriva:

Om man använder `\LaTeX`
`\footnote{uttalas`
`'lah-tekh'}` så
blir det bra. Alla rapporter
blir automatiskt snyggt
utformade.

Fotnoter numreras automatiskt 1,2,... Fast här blev "numret" på fotnoten "a" av olika anledningar. Observera att man skriver två apostrofer (' ') i stället för citationstecken (").

Om man använder \LaTeX^a så blir det bra. Alla rapporter blir automatiskt snyggt utformade.

^auttalas "lah-tekh"

Punktlistor är enkla:

<pre>\begin{itemize} \item första punkten \item här kommer den andra punkten i listan \end{itemize}</pre>	första punkten här kommer den andra punkten i listan
---	--

Numrerade listor är lika enkla:

<pre>\begin{enumerate} \item första punkten \item här kommer den andra punkten i listan \end{enumerate}</pre>	första punkten här kommer den andra punkten i listan
---	--

OBS! Pga hur detta dokument är formaterat så får listelementen inga punkter eller siffror, men i den vanliga dokumentklassen `{article}` blir det som förväntat.

Några klasser som vi använder:

```
\begin{description}
\item[SimpleWindow] Beskriver ett
enkelt ritfönster
\item[Scanner] Inläsning från
tangentbordet
\item[Random] Slumptal
\end{description}
```

Några klasser som vi använder:

```
SimpleWindow Beskriver ett enkelt
ritfönster
Scanner Inläsning från
tangentbordet
Random Slumptal
```

I dokumentklassen `article` blir det något annorlunda layout på definitioner. Använd en `tabular`-omgivning med kolumnspecifikationen `p{bredd}` för att få layout som liknar den ovan.

En tabell där den första kolumnen är vänsterinpassad (l), den andra centrerad (c) och den tredje högerinpassad (r). & avgränsar kolumnerna, \\ betyder ny rad, ~ är ett "hårt" blanktecken. \hline är ett streck.

```
\begin{tabular}{lcr}
  Produkt & Typ & Pris \\
\hline
  Skruvar & stora & 0.18 kr \\
  Muttrar & M16 & 0.38 kr \\
  Spikar & 12 tum & 0.12 kr \\
\end{tabular}
```

Produkt	Typ	Pris
Skruvar	stora	0.18 kr
Muttrar	M16	0.38 kr
Spikar	12 tum	0.12 kr

Flytande tabeller

Med en `\table`-omgivning skapar man en tabell med en förklarande text och ett nummer. \LaTeX placerar tabellen där det är lämpligt.

```
\begin{table}
\begin{tabular}{lcr}
  Produkt & Typ & Pris \\
\hline
  Skruvar & stora & 0.18 kr \\
  Muttrar & M16 & 0.38 kr \\
  Spikar & 12 tum & 0.12 kr \\
\end{tabular}
\caption{Våra produkter}
\end{table}
```

Produkt	Typ	Pris
Skruvar	stora	0.18 kr
Muttrar	M16	0.38 kr
Spikar	12 tum	0.12 kr

Tabell 7. Våra produkter

Att referera till etiketter

Om man sätter en etikett på en tabell kan man referera till den från texten. Exempel:

```
\begin{table}
\begin{tabular}{lcr}
  Produkt & Typ & Pris \\
  \hline
  Skruvar & stora & 0.18 kr \\
\end{tabular}
\caption{Våra produkter}
\label{produkter}
\end{table}
```

\\ Tabell 7. Våra produkter

Senare i texten: våra produkter
finns i tabell 7.

Senare i texten: våra produkter
finns i tabell~\ref{produkter}.

Figurer hanteras likadant som tabeller, i en `\figure`-omgivning.

För att infoga en programlista i en rapport använder man kommandot `\VerbatimInput{filnamn}` från paketet `fancyvrb`. Man bör *inte* använda "standard"-kommandot `\verbatiminput` eftersom det kommandot ignorerar alla tabulatortecken i programmet, och det medför att indragningarna försvinner.

```
\usepackage{fancyvrb}
\fvset{tabsize=4}
\fvset{fontsize=\small}
```

```
\VerbatimInput{Point.java}
```

```
class Point {
    private int x;
    private int y;

    public Point(int x, int y) {
        this.x = x;
        this.y = y;
    }
}
```

Öka eller minska avstånd

Ibland behöver man öka avståndet i vertikalled mellan två avsnitt i texten, till exempel före eller efter en tabell. Det kan man göra med kommandot `\vspace{längd}`, där längden kan anges i millimeter eller punkter eller något annat som \LaTeX känner igen. Längden kan vara negativ om man vill minska avståndet. Det finns också specialkommandon för att lägga in ett litet, mellanstort och stort avstånd:

```
\smallskip \medskip \bigskip
```

Man kan öka eller minska horisontellt avstånd med `\hspace{längd}`.

\LaTeX är *mycket* bra på att formatera matematisk text. Alla (tror jag) artiklar och böcker som innehåller matematiska formler är skrivna med \LaTeX . Man kan skriva formler antingen inuti löpande text eller på en egen rad:

I texten: formeln inleds med $\$$ och avslutas med $\$$.

På egen rad: formeln inleds med $\text{\backslashbegin{displaymath}}$ och avslutas med $\text{\backsend{displaymath}}$. $\text{\backbegin{equation}}$ och $\text{\backend{equation}}$ ger samma resultat men formeln numreras. Med \backlabel och \backref kan man etikettera och referera till ekvationer.

Enkla formler

Formeln $x=3y-2$ står
inne i texten. Däremot
står

```
\begin{displaymath}
```

$$x=3y-2$$

```
\end{displaymath}
```

för sig själv precis som

```
\begin{equation}
```

$$x=3y-2$$

```
\label{xochy}
```

```
\end{equation}
```

I ekvation~\ref{xochy} fann

vi att \ldots

Formeln $x = 3y - 2$ står inne i texten.

Däremot står

$$x = 3y - 2$$

för sig själv precis som

$$x = 3y - 2 \tag{1}$$

I ekvation 1 fann vi att ...


```
\begin{displaymath}
\alpha \leq \pi \approx 3.141592654
\end{displaymath}
```

$$\alpha \leq \pi \approx 3.141592654$$

```
\begin{displaymath}
x_{k+1} = x_k - f(x_k)/f'(x_k)
\end{displaymath}
```

$$x_{k+1} = x_k - f(x_k)/f'(x_k)$$

Exponenter, rötter

```
\begin{displaymath}
e^x = 1+x+x^2/2!+x^3/3!+\cdots
\end{displaymath}
```

$$e^x = 1 + x + x^2/2! + x^3/3! + \dots$$

```
\begin{displaymath}
x_{1,2}=\frac{p}{2}\pm
\sqrt{\frac{p^2}{4}-q}
\end{displaymath}
```

$$x_{1,2} = \frac{p}{2} \pm \sqrt{\frac{p^2}{4} - q}$$

Integraler, summor

```
\begin{displaymath}
\int_{-\infty}^{\infty}
e^{-x^2} dx
\end{displaymath}
```

$$\int_{-\infty}^{\infty} e^{-x^2} dx$$

```
\begin{displaymath}
\sum_{k=1}^n \frac{1}{a_k}
\end{displaymath}
```

$$\sum_{k=1}^n \frac{1}{a_k}$$

```
\begin{displaymath}
  \sin^2 x + \cos^2 x = 1
\end{displaymath}
```

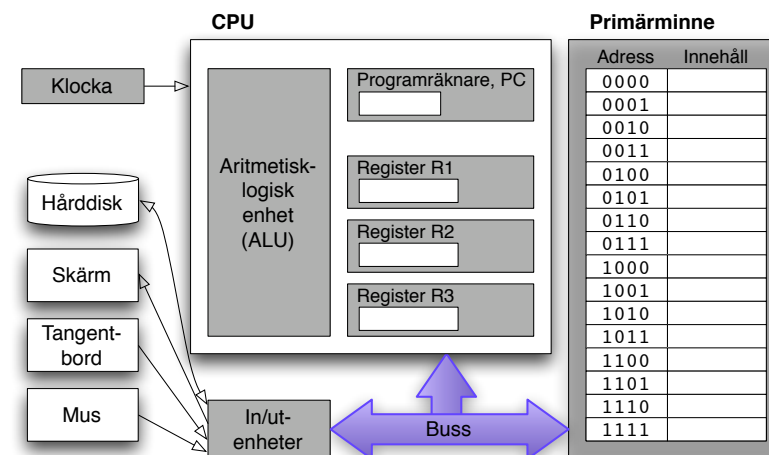
$$\sin^2 x + \cos^2 x = 1$$

Matriser, parenteser

```
\begin{displaymath}
A=\left(
\begin{array}{cccc}
a_{11} & a_{12} & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{n1} & a_{n2} & \cdots & a_{nn}
\end{array}
\right)
\end{displaymath}
```

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

Bilder kan inkluderas i \LaTeX -dokument om de är i formatet pdf, jpeg eller png (eps om man använder latex). Man måste använda paketet graphicx (eller graphics).



```
\usepackage{graphicx}
```

```
\includegraphics[height=40mm]{images/bild.pdf}
```

ImageMagick-programmet convert kan konvertera från och till de flesta bildformat:

```
convert bild.fig bild.pdf
```

Egna kommandon

Man kan lätt definiera egna kommandon, till exempel ett kortare namn för en text som man använder ofta. Kommandon kan ha parametrar.

```
\newcommand{\java}[1]  
{\texttt{#1}}
```

```
Klasser: \java{Random},  
\java{Scanner} och  
\java{PrintStream}.
```

```
Klasser: Random, Scanner och  
PrintStream.
```

Man kan definiera om existerande kommandon med `\renewcommand`. Det kan ställa till förvirring, så gör inte det.