

T_EX och L^AT_EX i desktop publishing

Anders Källström

01-11-20

Innehåll

1	Vad är T _E X och L ^A T _E X?	1
2	Att skriva text	2
2.1	Typsnitt	3
2.2	Miljöer	3
3	Att skriva matematik	4
4	Listor och tabeller	4
5	Grafik och färg	6
6	Lite blandade karameller	7

Detta är ett dokument för att visa några möjligheter i T_EX och L^AT_EX. Speciellt kommer vi att använda pdfL^AT_EX för att generera en pdf-fil direkt.

1 Vad är T_EX och L^AT_EX?

T_EX är ett typsättningsprogram som skapades i slutet 1970-talet av en matematiker och datavetare, Donald Knuth. Han skulle skriva en monografi, *The Art of Computer Programming* i flera delar. Han var emellertid missnöjd med kvalitén på slutliga produkten. Inte innehållet men utseendet. Matematisk typografi är svårt och det finns många och krångliga regler att hålla reda på. Han gjorde därför ett långt uppehåll i sitt skrivande och ägnade sig åt att ta fram ett datorprogram som kunde typsätta hans text med all den finess som han ville ha. Resultatet blev T_EX. Programmet är dock inte särskilt användarvänligt, inte ens med den måttstock som gällde på 70-talet. Därför skapades L^AT_EX av Leslie Lamport från DEC (Digital Equipment Corporation). T_EX är egentligen ett programmeringsspråk där vi kan tala om för en dator hur vi vill att en text ska typsättas. L^AT_EX är en stor samling av förberedda kommandon *makros* som i hög grad underlättar vårt skrivande. Man kan naturligtvis också skriva egna makros för att utföra speciella typsättningsuppgifter. Mer om detta senare.

De flesta människor som skriver olika texter använder idag en ordbehandlare. När man använder en sådan ägnar man stor möda åt att fixa layouten (ska jag ha indrag här och fetstil där . . .). I L^AT_EX är filosofin annorlunda. Det är ett *markup*-språk, dvs jag indikerar direkt i texten, med allmänna kommandon, hur jag vill att det ska se ut. Vill jag ha ett ord *kursiverat* så anger jag detta genom att skriva `\emph{kursiverat}` i texten

(ordet `emph` är en förkortning av engelskans "emphasize"). Om jag sedan vill att dessa ord istället ska vara i fetstil så kan jag ändra definitionen av kommandot `\emph`. Inget letande i texten för att hitta alla ställen som ska ändras. Det påminner om html eller ännu mera sgml.

2 Att skriva text

Att skriva text i \LaTeX är enkelt. Man gör det med en helt vanlig texteditor. Ett ordbehandlingsprogram går också bra, men man ska spara sina filer som en ren textfil, utan all formatering. I texten behöver man inte tänka på mellanslag och radbrytningar och liknande, allt detta tar \TeX hand om automatiskt. Så kommer t.ex. följande texter att typsättas lika

Detta är en text

och

Detta är en text

Det finns speciella kommandon för att skapa horisontella och vertikala mellanrum, `\hspace{1cm}` för ett 1cm horisontellt mellanrum och `\vspace{20pt}` för 20punkters vertikalt mellanrum. Med hjälp av dessa kan jag då skriva "Detta är en text" om jag skulle vilja.

En typisk \LaTeX -fil börjar med en *preamble*. För detta dokument ser den ut som följer

```
\documentclass[11pt,a4paper]{article} %allmänna inställningar
\usepackage[english,swedish]{babel} %svensk och engelsk avstavning
\usepackage[pdftex]{graphicx,color} %färg och grafik-makros
\usepackage[latin1]{inputenc} %tecken-encoding
\usepackage{fancybox} %box-makros
\usepackage{amsmath} %matte-makros
\usepackage{multicol} %flera kolumner
\usepackage{geometry} %sid-inställningar
\geometry{verbose,paper=a4paper,body={15cm,23cm},nohead}
\usepackage{pslatex} %postscript-fonter
\usepackage[pdftex,bookmarks=true,colorlinks=true,pdfstartview=FitH]{hyperref}

\title{\TeX\ och \LaTeX\ i desktop publishing}
\author{Anders Källström}
\date{01--11--20}
```

Om vi tittar på vad några av de olika raderna säger så är det

documentclass babel	ger allmänna inställningar. Här har vi valt den enkla typen article används för olika avstavningsregler. Vi har svenska som default, men kan enkelt byta till engelska
graphicx,color inputenc	laddar makros för att kunna använda grafik och färg väljer textencodingen. latin1 är den vanliga. Vill man skriva på ryska får man välja en annan.
geometry	beskriver sidan. Vi har valt en a4-sida med texten 15cm bred och 23cm hög utan header. Det finns en foot som innehåller sidnummer, men den är standard i article-klassen.
pslatex hyperref	ger mig Times, Helvetica och Courier som standardfonten gör inställningar till pdf-filen. Ger bl.a. innehållsförteckningen som klickbara länkar
title, author date	används för att göra en titel till dokumentet. Används även av hyperref för att ge information till pdf-filen

Sedan följer början på filen med raderna

```
\begin{document}
\maketitle
\tableofcontents
\vspace{2em}\noindent
```

Dessa kommandon har följande betydelse

<code>\begin{document}</code>	initierar början på mitt dokument. Det som kommer efter denna rad typsätts om det inte är ett \LaTeX -kommando
<code>\maketitle</code>	Gör en titel på dokumentet baserat på informationen i title/author/date-fälten
<code>\tableofcontents</code>	gör automatiskt en innehållsförteckning baserat på section/subsection...-rubrikerna
<code>\vspace{2em}</code>	vertikalt mellanrum 2em
<code>\noindent</code>	ingen indentering på nästa rad

Sedan kommer text...

2.1 Typsnitt

\TeX använde ursprungligen speciella typsnitt som kallades Computer Modern. De var ett slags variant av Times. Numera kan man använda vilka typsnitt som helst, men postscript är att föredra, i synnerhet om man vill generera pdf-filer. Typsnittshanteringen är dock fortfarande lite av ett problem. Det är ganska svårt att installera nya typsnitt. Det finns versioner av \TeX som haterar detta på bättre sätt, men de är alla kommersiella. Windows-versioner av \TeX

kan ofta använda även Truetype-fonter.

Som ni märker började jag här att skriva texten i två kolumner. Denna som jag behöver göra i textfilen (förutom att ladda paketet *multicol*) är att skriva `\begin{multicols}{2}` där jag vill börja skriva 2 kolumner och `\end{multicols}` där jag vill sluta. Allt annat sköter sig själv. Sådana saker som *gutter-space* kan ställas in. Kolumnerna balanseras i möjligaste mån automatiskt.

2.2 Miljöer

Det mesta som man skriver i \LaTeX skrivs i särskilda *miljöer* (eng. environments). Detta är ett slags mallar som formaterar text på ett i förväg angivet sätt. En miljö startas med

kommandot `\begin{miljö}` och avslutas med `\end{miljö}`. Man kan enkelt skriva egna sådana miljöer för att kontrollera utseendet på texten. Under årens lopp har vänliga människor skrivit otaliga sådana och donerat dem till \TeX -världen för att andra ska kunna använda dem. Allt som rör \TeX och \LaTeX finns att hämta gratis på Internet. Den som är intresserad kan gå till <http://www.tug.org> TeX User Group (detta är en klickbar länk) och hämta mer information.

3 Att skriva matematik

\TeX skapades för att skriva formler och matematik. Var och en som har behov av att skriva allehanda formler rekommenderas numera att använda \LaTeX . De stora förlagen i världen, Springer, Kluwer, Addison-Wesley, Wiley ..., har alla egna mallar för \TeX och \LaTeX för all sin produktion av vetenskapliga tidskrifter och böcker. Några exempel kanske är på sin plats. Alla som någon gång läst trigonometri har säkert stött på dessa funktioner $\sin \theta$, $\cos \omega$ eller varför inte

$$\tan \phi = \frac{\sin \phi}{\cos \phi}.$$

Den sista raden skrivs på följande sätt

```
\begin{displaymath}
\tan\phi = \frac{\sin\phi}{\cos\phi}.
\end{displaymath}
```

Vi använder miljön *displaymath* för att få en fristående rad med matematik. Ett bråk (eng fraction) skrivs som `\frac{täljare}{nämnare}`. Inom dtp-industrin har man kanske inte så stora behov av att skriva matematiska formler, men jag kan väl ändå ge några exempel på sådant som man enkelt skriver i \LaTeX

$$\det A = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix} = \sum_{\pi \in S_n} (-1)^{\sigma(\pi)} a_{1\pi(1)} a_{2\pi(2)} \cdots a_{n\pi(n)}$$

eller

$$\int_0^{\infty} \frac{x^{a-1}}{e^x - 1} dx = \Gamma(a)\zeta(a) \quad \text{för } \text{Re } a > 1$$

(Jag garanterar inte riktigheten av formlerna...)

Man kan med hjälp av andra paket skriva kemiska formler. Det finns också möjlighet att skriva musikpartitur (med många stämmor) som är en speciell typ av formelspråk.

4 Listor och tabeller

Många dokument innehåller punktlistor eller numrerade listor. Dessa skapas i \LaTeX med speciella environments. En punktlista görs med *itemize*-miljön. Så kan vi t.ex. skriva

```
\begin{itemize}
\item Detta är den första punkten
\item Detta är den andra punkten, där vi gör texten lite längre. Vi vill se hur en radbrytning ser ut i en punktlista. Det avstånd som vi ser mellan punkten (eng bullet) och texten kan enkelt ändras om så önskas.
\end{itemize}
```

Efter typsättning ser detta ut som följer

- Detta är den första punkten
- Detta är den andra punkten, där vi gör texten lite längre. Vi vill se hur en radbrytning ser ut i en punktlista. Det avstånd som vi ser mellan punkten (eng bullet) och texten kan enkelt ändras om så önskas.

En numrerad lista skrivs på samma sätt fast med miljön *enumerate*:

1. Punkt 1
2. Punkt 2
3. Punkt 3

Det går också att själv ange vad som ska användas i numreringen. Om du skriver

```
\begin{itemize}
\item[a] Detta är den första punkten
\item[b] Detta är den andra punkten, där vi gör texten lite längre. Vi vill se
hur en radbrytning ser ut i en punktlista. Det avstånd som vi ser mellan
punkten och texten kan enkelt ändras om så önskas.
\end{itemize}
```

så blir det

- a) Detta är den första punkten
- b) Detta är den andra punkten, där vi gör texten lite längre. Vi vill se hur en radbrytning ser ut i en punktlista. Det avstånd som vi ser mellan punkten och texten kan enkelt ändras om så önskas.

Det finns också en miljö *description*. Denna kan t.ex. användas på följande sätt:

```
\begin{description}
\item[itemize] ger en punktlista
\item[enumerate] ger en numrerad lista
\item[description] ger en ''förklarings-lista''
\end{description}
```

itemize ger en punktlista

enumerate ger en numrerad lista

description ger en "förklarings-lista"

Tabeller är lätta att skapa i \LaTeX . Miljön *tabular* används för att ordna material o kolumner. Det ser ut på följande sätt

```
\begin{center}
\begin{tabular}{rcl}
kolumn 1 & kolumn 2 & kolumn 3 \\
\end{tabular}
\end{center}
```

vilket blir följande

kolumn 1	kolumn 2	kolumn 3
detta	skrivs i	rad 2

Vi ser att den första kolumnen är högerjusterad, den andra centrerad och den tredje vänsterjusterad. Detta skrivs med `rcl` i `\begin{tabular}{rcl}`. Kolumnbredden anpassas till det som står i kolumnen. Det går också att ange en fix bredd på kolumnen.

Ofta vill man ha en del vertikala och horisontella linjer i en tabell för att avskilja kolumner och rader. De vertikala linjerna anges i kolumnformatet `r|cl` som t.ex. `r|cl` för att få en vertikal linje mellan kolumn1 och kolumn2. Horisontella linjer skapas på den plats i tabellen där de ska stå med `\hline`.

```
\begin{center}
\begin{tabular}{r|cl}
kolumn 1 & kolumn 2 & kolumn 3 \\
\hline
detta & skrivs i & rad 2
\end{tabular}
\end{center}
```

vilket ger

kolumn 1	kolumn 2	kolumn 3
detta	skrivs i	rad 2

eller kanske man vill ha

```
\begin{center}
\begin{tabular}{|rcl|}
\hline
kolumn 1 & kolumn 2 & kolumn 3 \\
detta & skrivs i & rad 2 \\
\hline
\end{tabular}
\end{center}
```

kolumn 1	kolumn 2	kolumn 3
detta	skrivs i	rad 2

5 Grafik och färg

För att lätta upp framställningen kan jag väl inkludera en bild kanske.



Den är gjord med programmet Gimp som är motsvarigheten till Photoshop i Linux. Konstnären är japan och heter Sakura. Bilden är infogad direkt i jpeg-format i \LaTeX -filen med kommandot

```
\begin{center}
\includegraphics[width=5cm]{sakura-tn.jpg}
\end{center}
```

Notera hur enkelt det är att ange hur stor man vill att bilden ska vara. I standard \TeX och \LaTeX använder man helst bilder i eps-format (encapsulated postscript). När man gör pdf-filer så kan man använda bilder i pdf, png eller jpeg-format.

Vad gäller färg så det lika enkelt att använda. Om man vill skriva **någon text i färg** så görs detta med kommandot `\textcolor{red}{någon text i färg}`. Det finns flera färgmodeller, bl.a. *cmyk*, *rgb* och *named* som jag använde i exemplet. En färg anges med ett namn (red som ger rött). Man kan också definiera egna färger, **här använder vi magenta**. Den färgade texten skrevs som

```
{\color[cmyk]{0,1,0,0} här använder vi magenta}
```

Notera hur klammrarna avgränsar den text som **typsätts i den valda färgen**.

6 Lite blandade karameller

Man kan, om man vill placera texten där man önskar med hjälp av boxar. Med följande kommando, `\raisebox{10pt}{blabla}`, kan man flytta upp texten blabla 10punkter från baslinjen, så här `\raisebox{10pt}{blabla}`. Notera att radavståndet automatiskt ökar för att ge plats åt boxen. Om man inte vill att det ska göras så kan man med `\smash{}` tala om att boxen inte har någon höjd. Då får man i stället `\smash{blabla}`. Med lite möda kan man alltså skriva texten på ett sätt så man blir sjösjuk, Anders Källström.

Ja detta blev ett långt dokument. Jag har ändå bara skrapat på ytan av vad man kan göra med \LaTeX .

Använd \LaTeX !