Typesetting with TEX / LATEX

Part VII: Useful Things...

F. C. Langbein

School of Computer Science
Cardiff University

Version 1.0

Overview

- Part I: basic components and essential LATEX
- Part II: formatting and layout
- Part III: figures and tables
- Part IV: basic mathematics and AMSLATEX
- Part V: PDFLATEX and slides
- Part VI: BIBTEX and MakeIndex
- Part VII: useful things...

TEX Directory Structure

- TDS is a standard for organising TEX files
- All positions relative to a root directory `TEXMF`
  - Usually `/usr/share/texmf`, `/usr/local/share/texmf` or `/opt/texmf`
- Basic structure under `TEXMF`:
  `<program>/package/...`
- Under Un*x a command to find files, directories, etc. exists:

```
> kpsewhich package.sty
> kpsewhich class.cls
> kpsewhich -expand-var='$TEXMF'
```

  - See man-page or use `--help` argument

Main TDS Directories

- Packages and classes stored under:
  - `TEXMF/tex/FORMAT/PACKAGE/`
  - e.g. `TEXMF/tex/latex/base/`, `TEXMF/tex/latex/graphics/`, ...
- Documentation files stored under:
  - `TEXMF/doc/FORMAT/PACKAGE`
  - e.g. `TEXMF/doc/latex/base/`, `TEXMF/doc/help/`, ...
- Other top-level directories:
  - `fonts`, `metafont`, `bibtex`, `metapost`, `source`

TEX Package Documentation

- TEX packages (classes, styles, etc.) usually come in a source file format (`.dtx`)
- Source files are a mixture of documentation and TEX sources
- For package, class files, etc. documentation is stripped from sources
- To get the actual documentation run latex on `.dtx` file:

```
> latex graphics.dtx
```

- Sources may be installed under
  `TEXMF/source/FORMAT/PACKAGE`
  (at similar place than stripped file under `TEXMF/tex`)
- Note, sources may contain more than one package file

Defining New Commands (TEX)

- TEX macros can be changed or new ones can be defined
- TEX syntax for changing commands:

```
\def#1#2{Definition}
```

  - `#1`, etc. specifies list of arguments
- Some examples:

```
\def\myname{F.~C.~Langbein}
\def\heading#1{\textbf{\large #1}}
```

- This is the TEX way of defining macros!

Defining New Commands (LaTeX)

- To define a new macro under LaTeXuse:

```
\newcommand{\NAME}[narg]{MACRO}
```

- Example:

```
\newcommand{\xvec}[1]{\ensuremath{#1_1,\ldots,#1_n}}
```

  - Defines new `\xvec` command:

```
$\xvec{x}$ and \xvec{y}
```

$x_1, \ldots, x_n$ and $y_1, \ldots, y_n$

- To redefine a macro using LaTeX style:

```
\renewcommand{\NAME}[narg]{MACRO}
```

- Under LaTeX this is the proper way to redefining commands, but note that `\def` also works (if you are not a purist)

Redefining Macros

- You may redefine macros provided by classes and packages
- For simple modifications:
  - Find the definition in the class/package files and rewrite it
  - It is often sufficient to get the basic idea of the command in order to make a simple adjustment
- Redefinition has to be after original definition (after the package inclusion)
- Sometimes this may mean you have to redefine a command with the `@` character in it
  - `\makeatletter` makes `@` a useable letter character for macro names
  - `\makeatother` switches `@` back to an "other" character

Redefining `\section`

- In `article.cls` you find the definition:

```
\newcommand\section{\@startsection {section}{1}{\z@}%
  {-3.5ex \@plus -1ex \@minus -.2ex}%
  {2.3ex \@plus.2ex}%
  {\normalfont\Large\bfseries}}
```

- `\@startsection` is the basic macro for creating sections
- For small-caps section:

```
\makeatletter
\newcommand\section{\@startsection {section}{1}{\z@}%
  {-2.5ex \@plus -1ex \@minus -.2ex}%
  {1.3ex \@plus.2ex}%
  {\normalfont\Large\bfseries\scshape}}
\makeatother
```


More on Defining Macros

- For more on defining macros and TeX internals start with reading **The TeXbook** by D. Knuth
- Also read the sources and package/class files...