#### Instructions for use with Acroread

#### Mouse activated controls:

quit Acroread Quit close document Close document info Info jump to this page Help go back previous page pop-up bookmarks **Bookmarks** pop-up thumbnails **Thumbnails** next page go forward internal link go to link

external link go to link via browser

Keyboard activated controls:

'page down'
'page up'
'cursor up'
'cursor down'
'enter'
'escape'

next page
previous page
previous bookmark
next bookmark
go to bookmark
window mode

# An Introduction to Type-setting projects in LATEX with the UoYCSProject class

Jeremy Jacob

2005 September 27



# What is LATEX?

LATEX is a document description language built on top of Donald Knuth's TEX type-setting engine.

Cf. HTML and SGML/XML applications.



#### A minimal document

Source

\documentclass { minimal }

\begin { document }

Hello\_World.

\**end**{document}

## Output

Hello World.





## Why use LATEX?

- The sophisticated type-setting algorithm of T<sub>E</sub>X, and the enhanced algorithm of pdfe(la)tex. (See the T<sub>E</sub>X showcase.)
- The huge number of pre-defined packages for doing common things. (See the TEX catalogue.)
- The ability to define your own special purpose structures.
- Stable basis.
- Good for large, academic documents.



#### References

There are many good references for TEX and friends.

See "A guide to type-setting project reports in LATEX with the UoYCSproject class".



## UoYCSProject — a class for project reports

There are many pre-defined document classes:

Base minimal, article, report, book, letter, slides.

**KOMA-Script** scrartcl, scrreprt, scrbook, scrlttr2.

Memorandum memorandum.

Others ..., beamer, ..., UoYCSproject, ...



## Text, commands and environments

```
A LATEX source is a mix of:
text Some text.,
commands \LaTeX, \frac{2}{3}, and
environments
    \begin{verse}
      APRIL is the cruellest month, breeding
       \\Lilacs out of the dead land, mixing
       \\...
    \end{verse}
                            Bookmarks Thumbnails
Quit | Close |
         Info | Help
```

# The anatomy of a LATEX source

```
\documentclass[class options]{class name}
preamble (definitions and declarations)
\begin{document} % this is a comment.
body
\end{document}
```



## The anatomy of a UoYCSproject preamble

```
\documentclass[citation_style]{UoYCSproject}
% Order of declarations does not matter.
\author{Anne_Student-Name}
\title {A_Solution_to_the_Problem_of_$P=NP$}
\date{30, February, 2000}
\supervisor{Prof. Z. Soporific}
MEng
\wordcount{2,345}\excludes{Appendix~\ref{sec:code}}
\dedication {To_My_Cat,_Jeoffery}
\abstract{The well known problem of $P=NP$, is explained, together
__with_its_significance_and_a_brief_history_of_attempts_to_solve
__ it .__An_ingenious_solution_is_presented.}
\begin { document }
\end{document}
```

A full list of declarations is given in AGtTSPRiLwtUC, Figure 7.1, P 46.



## Extra definitions and package loading

You can load extra packages and make your own definitions.

These go in a file with the same name as your main file, but extension 'ldf'. This is different to the way all other classes work. (I have implemented UoYCSproject in this way to ensure that packages are loaded in the correct order.)

Useful packages include: listings, graphics, graphicx, pdftricks, amsmath.





## The anatomy of the body

Front matter Title pages, abstract, contents, &c.

**Main matter** The text, divided into (parts,) chapters (, sections, subsections, subsections, paragraphs and subparagraphs).

Back matter Bibliography, appendices &c.



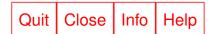
#### Front matter

Mostly generated automatically from the declarations. Optionally:

a list of figures generated by \listoffigures.

a list of tables generated by \listoftables.

document dependent lists for example, the 'listings' package provides \ Istlistoflistings .





#### Main matter

```
\part{ title }
                        % Optional
\chapter{ title }
                        % Compulsory
\section { title }
                        % Optional
\subsection { title } % Optional
\subsubsection{title} % Optional
\paragraph{title} % Optional
\subparagraph{title} % Optional
Text.
       Text.
                            Thumbnails
Quit Close
       Info | Help
                      Bookmarks
```

#### **Back matter**

```
\bibliography{file1,file2} % Construct bibliography
\appendix % remaining chapters are appendices
\chapter{title} % One per appendix
\section{title} % Optional
\subsection{ title } % Optional
\subsubsection{title} % Optional
\paragraph{title} % Optional
\subparagraph{title} % Optional
Text.
       Text.
                    Bookmarks Thumbnails
      Info Help
Quit | Close |
                                                Introduction to LATEX/14
```

#### Text elements

**Characters** Can control series, family, shape, colour and size of each text character. See AGtTSPRiLwtUC, §6.3.3.

Sentences Sentence\_one.\_\_Sentence\_two.

## **Paragraphs**

Paragraph\_one.\_\_\_%\_\_blank\_line\_separates\_paragraphs

Paragraph\_two.



## Special features

# **Context dependent emphasis**

Cross references Sectional units, floats, equations, &c.

**Quotations** Short and long

Lists Bulleted, numbered and labelled

**Tables** 

**Pictures** 

Floats Tables, Figures and others.



## Citations and the bibliography

Through the *natbib* package. Two style options:

**Harvard** Use option: authoryearcitations

Cite as text 'As Joyce (1939) says, life is complex.'

Cite in parenthesis 'Life is complex (Joyce, 1939).'

**Toronto** Use option: numericalcitations

Cite as number 'As Joyce [17] says, life is complex.' or 'Life is complex [17].'

Citations are kept in a database in a flat file and processed by a program called BibTEX before inclusion in output file.



#### **Mathematics**

Very powerful facilities. May be enhanced by amsmath packages.

**Inline** Here is a formula:  $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$ ; isn't it beautiful?

**Displayed** Here is a formula:

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2} \tag{1}$$

Isn't it beautiful?

#### **Definitions**

A major reason for using LATEX. Create special-purpose commands and environments for the structures in *your* document.

To define a command called \UoY that prints 'The University of York':

\newcommand\*{\UoY}{The University of York}

To define a command that has two parameters:

\newcommand\*{\C}[2]{\_{#1}C^{#2}}

\**begin**{math}\C{x+2}{3y}\**end**{math} type-sets as  $_{x+2}C^{3y}$ .

Quit | Close | Info Help





Introduction to LATEX/19

## Case study: cryptographic protocols — 1

A *message* has three components: *sender*, *receiver* and *content*. So we write our document in terms of a command \msg that has 3 parameters.

Two possible definitions:

\newcommand\*{\msg}[3]{#1\rightarrow#2:#3}

\newcommand\*{\msg}[3]{#2\Longleftarrow\left[#3\right]\Longleftarrow#1}

\msg{S}{R}{C^{A^{B}}} type-sets as either:

$$S \to R : C^{A^B} \text{ or } R \longleftarrow \left[C^{A^B}\right] \longleftarrow S.$$

Quit Close Info Help



Introduction to LATEX/20

## Case study: cryptographic protocols — 2a

A protocol is a sequence of messages. So we write our document in terms of an environment that collects a sequence of messages.

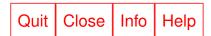
We will write, for example:

## Case study: cryptographic protocols — 2b

Now we design the printed form.

The output should have numbered messages to which labels can be attached. Each message should be printed on a line of its own.

The definitions of \msg and \sep should be local to the environment.





### Case study: cryptographic protocols — 2c

```
\newcounter{msgnumber}
\newenvironment * { protocol }
 {\setcounter{msgnumber}{0}%
  \newcommand * { \ msg } [ 3 ] { %
     \ refstepcounter { msgnumber } \ themsgnumber&##1&##2&##3}
  \newcommand * { \ sep } { \ \ }
  \begin { math } \ displaystyle%
     \begin{array}{r@{.\quad}|@{\rightarrow}|@{\;:\;}|}
 {\end{array}\end{math}}
                        Bookmarks | Thumbnails
Quit | Close |
       Info | Help
                                                        Introduction to LATEX/23
```

## Case study: cryptographic protocols — 2d

#### Source

Output

- 1.  $A \rightarrow B : X, Y, Z$
- 2.  $B \rightarrow C : W, X$
- 3.  $C \rightarrow B : W, X'$

\end{ protocol }



## How to run $\Delta T_E X - 1$

- 1. Create <source>.tex, <source>.ldf, bibliographic files, &c.
- 2. Run PDF(E)LATEX (Using TeXLive on Departmental Linux: 'pdflatex <source>'). Collects auxiliary information in <source>.aux, <source>.toc, &c. and creates <source>.pdf.
- 3. Run BibTEX ('bibtex <source>'). This uses the auxiliary information to determine database files and writes <source>.bbl file.
- 4. Run PDF(E)LATEX ('pdflatex <source>'). Collects auxiliary information in <source>.aux, <source>.toc, &c., including bibliographic cross references.
- 5. Run PDF(E)LATEX ('pdflatex <source>'). There should now be enough auxiliary information to generate the final version of <source>.pdf.



# How to run $\angle AT_EX - 2$

Process can be eased by tools such as

- AUCTeX package for emacs (Linux and Microsoft).
- MikTEX and WinEDT on Microsoft systems.

Incremental processing and errors do not mean repeating the whole process: for example, BibTEX only needs to be re-run if the bibliographic files change or a new citation is added.

