Document preparation in Unix

- TEXand LATEX
- graphviz
- xfig
- xv
- spell checkers
- printing

Word Processors

Word processors, such as Microsoft's Word® and OpenOffice's Writer, use the WYSIWYG model:

- ☞ Word processors are interactive.
- ☞ Word processors are relatively easier to learn
- ☞ Word processors are very useful for those who need to do simple documents occasionally.

Text formatters

Text formatters, such as TEX/LATEX, use the model of "markup", where text is decorated with markup commands and then processed by a program; output can then be viewed.

Characteristics, then, of text formatting:

☞ It tends to be batch-oriented

☞ Generally better control over the output

- ☞ Output generally looks better
- ☞ Much better for creating longer documents
- ☞ Much better for creating long-life documents
- ☞ Much better for creating series of related documents
- ☞ Having the source in text means that other text tools can be applied to the source.

TEXand LATEX

TEXwas invented in the late 1970s by Donald Knuth. The first generally useful release was probably TeX82 in 1982, though the language wasn't frozen until 1989.

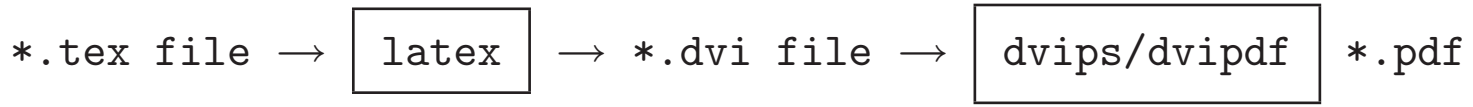
It was created to make nice mathematical documents, with emphasis on mathematical fonts since many of the easily available ones for electronic production were not high quality.

LATEXwas invented in 1985 by Leslie Lamport. It

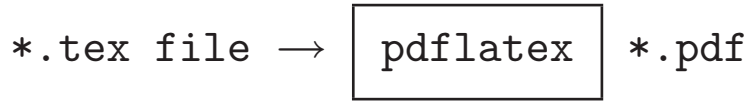
contains higher level support for many constructions such as table of contents, citations, floating tables and figures, and so forth.

Generating a LaTeX document

There are a variety of ways these days to generate a LaTeX document. The most general one is



The simplest these days combines these two steps:



The idea behind `dvi` files is that they were to be "device independent", and then output would go to a special driver for whatever output device might be available, such as our ancient Imagen printers.

Of course, Adobe invented PostScript® which instituted what was to become an equally device independent mechanism, at least to the level of fonts. The "Portable Document Format" (pdf) then added fonts to the output format. This was a bit of a muddle for TEXsince its model was to create its own fonts with the program Metafont, but these days, TEXalso can read and use other font families seamlessly.

Metafont and MetaPost

Fonts are created by the Metafont program, and graphics can be created with MetaPost.

Generally, you won't have to worry about this; LaTeXwill usually call Metafont seamlessly if it needs to recreate a font.

LaTeXcommands

A LaTeXfile must contain not only text but also markup commands. Commands consist of a special single characters or a words preceded by the backslash.

`%`	indicates a comment	`~`	represents a space
`&`	is used in making tables	`$`	is used to indicate math
`{`	starts an argument list	`}`	ends an argument list
`_`	precedes a subscript	`^`	precedes a superscript
`#`	used in defining commands		

Generally, these can be printed by preceding them with a backslash, though the safest thing is to use `SPECIAL`.

LaTeX comments

A comment begins with % and ends with the line.

This is similar to the C++ // or Ada -- comment.

Document structure with the "Article" class

```
\documentclass[12pt]{article}          % specify class
\usepackage{fancyvrb}                  % preamble: use a package
\usepackage{graphics}                  % preamble: use a package
\begin{document}                       % start the actual document to layout
\title{}                               % title of the article
\author{}                              % author of the article
\date{\today}                          % you can specify a date, or use today's
\maketitle                             % this displays the preceding
\tableofcontents                       % creates a table of contents
\begin{abstract}                       % start an abstract environment
\end{abstract}                         % end an abstract environment
\section{NAME}\label{}                 % start a section, create a label for it
```

```
\section{NAME}\label{}          % another section
\bibliography{}                 % generate a bibliography
\end{document}                  % finish the document
```

LaTeX document class

The document class defines the way that the document will be formatted.

Popular classes include:

```
article     % short articles such as journal papers
report      % longer works broken into chapters
book        % has chapters, treats odd and even pages differently
slides      % a slide package
foils       % another slide package
letter      % used for writing letters
exam        % used for making exams
```

For instance, to specify an article with an 11 point font, use

```
\documentclass[11pt]{article}
```

LaTeX packages

TeX is a Turing-complete language, and numerous packages have been created to support use of TeX and LaTeX.

You can access these packages with `\usepackage{}`.

For example,

```
\usepackage{graphics}
\usepackage{graphicx}
```

Beginning the document

To end the preamble and actually start creating displayable material (i.e., the "body" of your document), you insert the `\begin{document}` command; to end the document, you use `\end{document}`.

Environments

Environments allow you to specially treat text that environment uniformly. For instance, you might want to enumerate some items. Rather than having to write spacing and enumeration data for each item, you simply point what the items are:

```
\begin{enumerate}
\item This is item 1.
\item This is item 2.
\end{enumerate}
```

The LaTeX article heading

The LaTeX article header consists of the title, author, and date.

The `\title{TITLE TEXT}` command is used to store the text for the title.

The `\author{AUTHORS}` command is used to store the author information. You can use `\and` to separate multiple authors.

The `\date` command contains the date of the article.

If not specified, the current date will be used.

The LaTeX article heading, cont'd

The `\maketitle` command causes the title, author, and date information to be typeset into the article.

Depending on the style, the title might appear on its own page, or on the first page.

For example,

```
\title{Introduction to \LaTeX}
\author{John Doe \\
Florida State University}
\date{October 10, 2006}
\maketitle
```

Document spacing

The Wikipedia has a good description of TEX's input process at http://en.wikipedia.org/TeX. Here's a summary:

```
The system can be divided into four levels: in the first, characters
are read from the input file and assigned a category code (sometimes
called catcode, for short). Combinations of a backslash (really: any
character of category zero) followed by letters (characters of
category 11) or a single other character are replaced by a control
sequence token. In this sense this stage is like lexical analysis,
although it does not form numbers from digits. In the next stage,
expandable control sequences (such as conditionals or defined macros)
```

are replaced by their replacement text. The input for the third stage is then a stream of characters (including ones with special meaning) and unexpandable control sequences (typically assignments and visual commands). Here characters get assembled into a paragraph. TeX's paragraph breaking algorithm works by optimizing breakpoints over the whole paragraph. The fourth stage breaks the vertical list of lines and other material into pages.

Document spacing

In addition to simple paragraph breaking and setting in pages, LaTeXhandles *floating* figures and tables quite well.

Whitespace in the form of blanks and newlines indicate the end of a word. Otherwise it isn't significant.

New paragraphs can be indicated by at least one blank line.

LaTeX abstract environment

Abstracts are created in LaTeX with the `abstract` environment.

Example:

```
\begin{abstract}
This paper goes over the basics of \LaTeX.
\end{abstract}
```

LaTeX sectioning

A LaTeX article is divided with the following commands:

```
\section{NAME}
\subsection{NAME}
\subsubsection{NAME}
```

Section numbers and titles are saved for a table of contents if requested.

For example:

```
\section{The Art of \LaTeX}
```

```
\subsection{\LaTeX's Picture Environment}
\section{Font Fun in \LaTeX }
```

Labels and References in LaTeX

Sections are often referred to by number within a document. However, writers can and do decide to reorder sections. LaTeXallows writers to give internal names to sections, and then to refer to those names to avoid having to renumber internal references inside of documents.

For example:

```
\section{The Paucity of Comment Markers}
\label{paucity}
...
As mentioned in section \ref{paucity}, there are no suitable replacements...
```