

The ulem package: underlining for emphasis

Donald Arseneau
asnd@triumf.ca

2010/01/21

Abstract

The ulem package provides various types of underlining that can stretch between words and be broken across lines. Use it with L^AT_EX or plain T_EX.

In L^AT_EX, ulem replaces italics with underlining in text emphasized by `\emph`, and to some extent by `\em`. A declaration of `\normalem` or the `\usepackage` option `[normalem]` disables this feature.

The following commands are defined for general use:

<code>\uline{important}</code>	underlined text like <u>important</u>
<code>\uuline{urgent}</code>	double-underlined text like <u><u>urgent</u></u>
<code>\uwave{boat}</code>	wavy underline like <u>boat</u>
<code>\sout{wrong}</code>	line drawn through word like wrong
<code>\xout{removed}</code>	marked over like removed
<code>\dashuline{dashing}</code>	dashed underline like <u>dashing</u>
<code>\dotuline{dotty}</code>	dotted underline like <u>dotty</u>

Other similar commands can be defined with relative ease by utilizing the `\markoverwith` command provided by ulem.

The ulem package is Copyright © 1989–2010 by Donald Arseneau (Vancouver, Canada). The package (ulem.sty) and this documentation (ulem.ltx, ulem.pdf) may be freely transmitted, reproduced, or modified for any purpose provided that the copyright notice is left intact. (Small excerpts may of course be taken and used without any restriction.)

1 Basic Use

Ulem is a package for L^AT_EX or plain T_EX which provides various types of underlining that can stretch between words and be broken across lines. Such underlining is given by the `\uline` command, leaving the original `\underline` command available for math mode. For underlining in plain T_EX, use `'\input ulem.sty'`.

In L^AT_EX ulem replaces italics with underlining in text emphasized by `\em` or `\emph` – but only if the text is delimited by braces. Unlike regular `\emph` emphasis, nested ulem emphasis generates multiple underlining; it does not alternate on and off. To use `\uline` for underlining, but have `\em` and `\emph` still produce normal italics, load ulem with `\usepackage[normalem]{ulem}`, or declare `\normalem` in the preamble.

Unlike regular underlining, ulem allows line breaks, and manual hyphenation, within the underlined text; but it is far from perfect. It is most suitable for simple text like L^AT_EX: A document preparation system that may need to be underlined in a manuscript submitted for publication. Again, ulem can only give underlined text for `\em` when the text is delimited by explicit braces.

The thickness of the underline rule is given by the command macro `\ULthickness`; use `\renewcommand` or `\def` (not the usual `\setlength`) to change it. The depth of the underline is controlled by the length `\ULdepth`. The default value is a special flag (`\maxdimen`) which lets the current font control the depth. You can set a particular value to `\ULdepth` (using `\setlength`) to force a particular depth, either locally for a special purpose, or for the document as a whole. (See the definition of `\sout` for an example.)

Other types of underlining are defined as well: a wavy underline with `\uwave` (under-wave), double underlines using `\uuline` (two lines under this), dashed `\dashuline` (dashes underneath) or dotted `\dotuline` (dots below) underlines. Non-underlines are: a line to strike out text `\sout` (~~strike out~~), and text crossed-out with hatching `\xout` (~~cross out~~). See them tabulated in the abstract.

Alternative package: soul.

2 Defining new commands

You can define your own styles of overprinting or underlining by using the `\markoverwith` command in the definition of your new command. The definition should be something like:

```
\newcommand\cmd{\bgroup \markoverwith{\langle something \rangle}\ULon}
```

The ‘*something*’ can be as simple as a single character, or as complex as you can keep track of; it will likely contain some repositioning commands, perhaps `\raisebox`.

Producing a colored underline or strike-through is not supported, but it is quite easy to colorize using the `\markoverwith` mechanism: just put `\textcolor{...}` in the *something*, such as this definition:

```
\newcommand\reduline{\bgroup\markoverwith
  {\textcolor{red}{\rule[-0.5ex]{2pt}{0.4pt}}}\ULon}
```

If you really feel the need to make a new command with a truly flexible rule, then look in `ulem.sty` and copy from the definitions of `\uline` and `\sout`.

Any type of underlining can be substituted for any font-selection command by issuing a proper `\useunder` declaration:

```
\useunder{\underlinecommand}{\fontdeclaration}{\fontcommand}
```

e.g., `\useunder{\uuline}{\bfseries}{\textbf}` gives a double underline instead of bold face in \LaTeX .

The commands `\normalem` and `\ULforem` respectively disable and enable underlining for `\em/\emph`, and so do the `\usepackage` options `[normalem]` and `[ULforem]`. There is also the `\usepackage` option `[UWforbf]` to replace boldface with a wavy underline. These features use the `\useunder` command internally. `UWforbf` does handle bold in math mode, in a limited way, but it doesn't work in section titles, unfortunately, because the titles are not delimited by explicit braces when printed by the `\section` command. Currently under `UWforbf` the `\bfseries` command still produces bold face, but `\bf` makes an under-wave.¹ In plain \TeX there is no `\textbf` so you could say `\useunder{\UWave}{\bf}{}`.

¹To get under-waved section titles (in ordinary \LaTeX classes) you could define:
`\renewcommand\@secntformat[1]{\uwave{\csname the#1\endcsname}\hskip1em}`
and later specify `\section[...]{\uwave{...}}`. But you don't want to enter that swamp.

Some commands, such as `\` and `\hskip` are given special treatment to work within uline, but others (like `\marginpar`) are not. Support for others can sometimes be added by assigning special meanings in the token register `\UL@hook`. (In \LaTeX do `\addto@hook\UL@hook{\let\cmd\ULversion}`.) The UL versions of commands should be modelled on `\UL@hskip` or `\UL@cr`, and should include the test `'\ifx\ \LA@space'`. For example, support for `\marginpar` is added through the hook mechanism.

All the underlining commands are robust (self-protecting) in \LaTeX .

3 Complications

The various underlining commands are essentially textual, and will not work quite the same in math mode. But since some font commands, in the old- \LaTeX style ('oldfont') serve both for text and math, math mode is handled (in an approximate way). Generally, you should avoid using ulem's commands within math, but math may appear in the text argument to ulem's commands.

Every word is typeset in an underlined box, so automatic hyphenation is disabled, but explicit discretionary hyphens (`\-`) will still be obeyed. Several text-formatting commands are specially supported within the underlining: `\-`, `\`, `~`, `\`, `\newline`, `\linebreak`, `\nolinebreak`, `\penalty`, `\hskip`, `\hspace`, `\hfil`, `\hfill`, `\hss`. Displayed math is not supported. The special commands do have a problem: they end a group so any local assignments are lost.

The underlines continue between words, and stretch just like ordinary spaces do. Since spaces delimit words, there may be some difficulty with syntactical spaces (e.g. '2.3 pt'). Some effort is made to handle such cases, but sometimes (such as `\let\x= y`) the space is interpreted incorrectly. You can usually solve the problem by enclosing the offending command in braces or in a macro (like `\newcommand\xeqy{\let\x= y}`), but...

One important incompatibility with braces and macro replacement: **All the text in braces or coming from a macro is typeset in a box** (as if in `\mbox`). Consequently, braces will suppress stretching and line-breaking in the text they enclose. Moreover, the specially-handled commands `\-`, `\`, `\newline` and `\linebreak` are usually ignored if they appear inside extra braces. They operate only when the braces delimit a command parameter without introducing a level of grouping. (Even though braces delimiting

command parameters do not normally imply grouping, many commands will add their own grouping.) Thus, you should try to limit inner braces to short bits of text or for delimiting parameters to commands. For emergency repairs, see the sadistic ‘Marat/Sade’ example below. Syntactical spaces inside braces never cause a problem, nor do spaces in math mode.

Text produced by expansion of a command (macro) is boxed too, but `\`, `\` , and `\-` still work properly in the expansion text so that:

```
\newcommand\iff{if and only if} {\em \iff}
```

does not allow any stretching or linebreaking between words, but

```
\newcommand\iff{if\ and\ only\ if} {\em \iff}
```

allows stretching and linebreaking. There is a problem though: the `\` (backslash-space) between words closes a group and any local assignments will be lost, in particular, font changes and color changes.

This loss of local assignments will break some other standard commands, (e.g., `\cite`) which produce multiple ‘words’ using local assignments. The way to protect such commands is to bury them in an `\mbox`: `\emph{every\one agrees~\mbox{\cite{you,me}}.}`

With `ULforem` in effect, nested `\em` or `\emph` commands produce multiple underlining, but heed the warnings about braces above. To get italics without underlining, use `\it`, `\itshape`, or `\textit`. Nesting of other types of underline is also possible, but the ‘underlines’ may overlap.

Here is a simple example (highlighting all invented words):

<p>'Twas <code>\emph{brillig}</code> and the <code>\emph{slithy~toves}</code> did <code>\emph{gyre}</code> and <code>\emph{gim\-ble}</code> in the <code>\emph{wabe,\ }</code> All <code>\emph{mim\sey}</code> were the <code>\emph{boro\goves}</code> and the <code>\emph{mome raths outgrabe}</code>.</p>	<p>'Twas <u>brillig</u> and the <u>slithy toves</u> did <u>gyre</u> and <u>gimble</u> in the <u>wabe</u>, All <u>mimsey</u> were the <u>boro-</u> <u>goves</u> and the <u>mome raths</u> <u>outgrabe</u>.</p>
--	--

Note use of explicit hyphenation in ‘gimble’, the tie (~) that prevents a line break in ‘slithy toves’, but stretches like a usual space, and the regular linebreak in ‘mome raths outgrabe’.

Here is an ugly example showing how nested uline (`\emph`) needs to be broken up to allow line-breaks

No, I did `{\em not}` act in
the movie `{\em \emph{The}`
`\emph{Persecution} \emph{and}`
`\emph{Assassin}\-\emph{ation}`
`\emph{of} \emph{Jean-Paul}`
`\emph{Marat}`, as Per`\-`formed
by the Inmates of the Asylum of
Charenton Under the Direc`\-`tion
of the Marquis de`~`Sade!} But I
`{\em did}` see it.

No, I did not act in the movie
The Persecution and Assassination
of Jean-Paul Marat, as
Performed by the Inmates of
the Asylum of Charenton Under
the Direction of the Marquis
de Sade! But I did see it.

In the nested emphasis, `\emph` had to be given for each word separately so the spaces between could stretch and break into lines. Even the discretionary hyphen (`\-`) in ‘Assassination’ had to be outside the braces, but the hyphen in ‘Direction’ was just fine because it was not in nested braces. The same applies to other special commands like `\` and `~`. Also, the spaces are printed with only a single underline because they are outside the nested `\emph` commands. This example really illustrates that `ulem` does not handle nested emphasis very well! Nevertheless, it is fine for simple things.