L'extension ulem *
Soulignement pour marquer l'importance

Donald Arseneau
asnd@triumf.ca

18/03/2011

Résumé

✖The ulem package provides various types of underlining that can stretch between words and be broken across lines. Use it with LaTeX or plain TeX.

In LaTeX, ulem normally 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 :

`\uline{important}`	underlined text like important
`\uuline{urgent}`	double-underlined text like urgent
`\uwave{boat}`	wavy underline like boat
`\sout{wrong}`	line struck through word like ~~wrong~~
`\xout{removed}`	marked over like removed
`\dashuline{dashing}`	dashed underline like dashing
`\dotuline{dotty}`	dotted underline like dotty

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

*Ce fichier a pour numéro de version et a été mis à jour le 01/01/1900. Son titre original est « The ulem package: underlining for emphasis ».

1 Basic Use

✖Ulem is a package for LaTeX or plain TeX 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. To load this package in plain TeX, use '`\input ulem.sty`'.

In LaTeX 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 LaTeX : 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` (not the usual `\setlength`) to change it.[1] The depth of the underline is controlled by the length `\ULdepth`. The default value is a special flag (`\maxdimen`) which lets the depth vary depending on the current font. 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.)

1. Users of plain TeX should use \def wherever these instructions recommend \(re)newcommand, and dimen assignments wherever \setlength is mentioned.

Other types of underlining are defined as well : a wavey 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{`⟨*something*⟩`}\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 by regular `\uline` or `\sout`, 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 wavey 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[2] (if `\bf` is defined at all). In plain TEX there is `\bf` but no `\textbf` so you could say `\useunder{\UWave}{\bf}{}`.

Some commands, such as `\\` and `\hskip` are given special treatment to work within uline, but others 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\`*UL-version-of-cmd*`}`.) 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 ('oldlfont') 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.

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

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 while

```
\newcommand\iff{if and only if} ... \uline{\iff}
```

prevents stretching and line-breaking between words, the alternative

```
\newcommand\iff{if\ and\ only\ if} ... \uline{\iff}
```

allows stretching and line-breaking. There is a remaining 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) :

```
'Twas \emph{brillig} and the
\emph{slithy~toves} did \emph{gyre}
and \emph{gim\-ble} in the
\emph{wabe,\\ } All \emph{mim\-sey}
were the \emph{boro\-goves} and the
\emph{mome raths outgrabe}.
```

'Twas *brillig* and the *slithy toves* did *gyre* and *gimble* in the *wabe,*
All *mimsey* were the *borogoves* and the *mome raths outgrabe*.

Note use of explicit hyphenation in 'gimble' and 'borogoves', the tie (`~`) that prevents a line break in 'slithy toves', but stretches like a usual space, the '`\\`' that gives a proper linebreak, and the regular (unforced) 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! It should be reserved for simpler cases where it performs

well without effort.