Linux Symposium Sample Paper

Hints for authors and reviewers

John W. Lockhart
Red Hat, Inc.
ols-sponsors@fedoraproject.org

Robyn Bergeron
robyn.bergeron@gmail.com

Ralph Siemsen
ralphs@netwinder.org

Abstract

This sample paper contains tips and tricks to ensure that what you write is what appears in the *Proceedings* with as little editing as possible. The most important tips appear in Section 4—mandatory reading for all!

If you are new to LaTeX, please read this paper in its entirety, in particular the introductory Section 2. Viewing the source (`example/example.tex`) side by side with the PDF is recommended. There are a number of other illustrative `.tex` files in the `example` directory as well.

1 New this year

This document has been revised for 2011. Please direct all feedback to Ralph. Many thanks to John, Robyn and other contributors.

- As in previous years, papers are being managed in a source control system. The location and details have changed for 2011, please see Figure 1 for further information.
- The build process has been simplified to running `make` in the directory containing your paper. Many text editors can automate this process, and will take you directly to the problematic line(s) should any error(s) occur.
- The reference system for the Proceedings is now Fedora 14, with the `texlive-latex` and `subversion` packages installed.

2 LaTeX basics

This section gives a quick introduction for those new to LaTeX. Regular users may skip ahead to Section 3. A LaTeX document consists of one or more "source" files having the extension `.tex`. Any text editor can be used to create and modify these files.

The source file consists primarily of the text of your paper. Ordinary words, numbers and punctuation symbols require no special markup. Spacing between the words does not matter, nor do the line breaks. A blank line is used to separate paragraphs.

2.1 Special symbols

Several symbols have special meaning in LaTeX. Of particular note are the the percent symbol (%), the dollar sign ($), and the underscore character (_). A complete list appears in Table 1. To use any of these characters in your paper, you must escape them using a backslash (\).

Char	Command	Otherwise
#	`\#`	argument number
$	`\$`	toggle math mode
%	`\%`	comment: ignore rest of line
&	`\&`	tabstop
_	`\_`	subscript in math mode
{	`\{`	open environment
}	`\}`	close environment
~	`\~{}`	non-breaking space
~	`\textasciitilde`	non-breaking space
\	`\textbackslash`	begin command

Table 1: LaTeX characters that require special handling

To access the Subversion repository, you must provide a SSH public key. If you do not already have one, then run the `ssh-keygen` command to generate one.

```
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ralphs/.ssh/id_rsa): <Enter>
Created directory '/home/ralphs/.ssh'.
Enter passphrase (empty for no passphrase): <Enter>
Enter same passphrase again: <Enter>
Your identification has been saved in /home/ralphs/.ssh/id_rsa.
Your public key has been saved in /home/ralphs/.ssh/id_rsa.pub.
```

Send a copy of the public key (`id_rsa.pub`) to `ralphs` at `netwinder.org`. Keep the private key (the one without the `.pub` suffix) secure—do not give it to anyone, and do not e-mail it.
You can send the public key as an attachment, or simply paste it into your e-mail. Also please include the name of your paper, the name(s) of the author(s), and an abstract. In case you have already started working on the paper itself (perhaps using the 2009 template), then please include your source files.
You can expect a response within a day or two at the most. I will create/import your paper into the source control system, and will send back details on how to access your paper in the repository. In the meantime, you can begin work using the 2012 template, available at http://www.linuxsymposium.org/2012/guide.php.

Figure 1: Subversion access for Linux Symposium Papers

Quotation marks, both 'single' and "double," look good in body text, while other `"styles"` might look better for other uses. When typesetting for a compiler, punctuation goes outside the `"quotation marks",` but punctuation is placed *inside* the quotation marks for "narrative."

There are multiple flavors of dashes—the em dash, the en–dash, the oft-used hyphen, and the minus sign (math mode: $2x-3$). Note that the preceding sentence contains them all.

2.2 Font styles

Several different font styles are available. In addition to the commands given here, there are also several Linux Symposium-specific formatting commands for identifiers (variables, function names) which are described in Section 3.3.

- `texttt` produces `typewriter` style.
- `textit` produces *italics*.
- `textbf` produces **boldface**.
- `textsc` produces SMALL CAPS.
- `Font` **STYLES** can be ***combined***

Font styles should be used sparingly, otherwise the paper becomes quite difficult to read. Do use italics when introducing a new *term* for the first time, rather than using "quotes" each time.

2.3 Sectioning and lists

The LaTeX commands for breaking a document into sections and subsections are `\section{}` and `\subsection{}`, respectively. Section numbering is automatic, and may be suppressed by using "starred" version of the command.

Several list styles are supported: `enumerate` produces numbered list items, while `itemize` produces bullets. Lists can be nested if desired. Figure 2 shows the typical usage.

2.4 Figures and tables

Use of figures, tables and graphics can improve clarity of your paper. Any such elements should placed into a "float", so that LaTeX can optimize the page layout. If this is not done, then the elements are forced to appear

```
% A list with numbered items
\begin{enumerate}
\item This is the first item.
\item This is the second item.
Still part of the second item.
\end{enumerate}
```

Figure 2: How to create lists

at a particular location, and this results in a lot of empty space, awkward line-breaks, or both.

Since these elements can "float" around, each one should be given a unique identifier using the `\label{}` command. To refer to the floating element from elsewhere in your document, use the `\ref{}` command with the same identifier[1].

To incorporate graphics into your document, use the `\includegraphics{}` command. Please use only the base filename (no directory or path names), and also omit the file extension. Section 3.1 contains many more details about graphics formats.

For cosmetic reasons, a small margin should be left around each figure. Figure 3 shows typical examples of full-width or full-column picture insertion.

Note that LaTeX makes a distinction between `figure` and `table`, and maintains separate numbering for each. This distinction is unfortunately lost on many readers and authors. If this applies to you, just use `figure` everywhere, and all will be well.

2.5 Further LaTeX resources

If you need further help with LaTeX, there are many sources of information available. A good starting point is http://en.wikibooks.org/wiki/LaTeX, and also the beginners guide at http://www.tug.org/.

Some recommended books include *A Guide to LaTeX* (Kopka & Daly, ISBN 0-201-39825-7) and the *LaTeX Graphics Companion* (Goossens, Rahtz, & Mittelbach).

Also note that your machine likely includes a lot of documentation. Run the command `info latex` or look in `/usr/share/doc`.

[1] This is not limited to figures; labels can appear anywhere in the document. Thus you can easily refer to another section.

3 Linux Symposium particulars

While the previous section was aimed at newcomers to LaTeX, this section covers issues applicable to all Linux Symposium paper authors.

3.1 Graphics and Images

Bit-mapped graphics must be avoided as they do not scale well, resulting in poor quality of printed copies. Avoid using tools such as PhotoShop, PowerPoint, or the GIMP. The only acceptable use of bitmaps is for screen-shots or photographs—please provide high quality originals, rather than scaled-down versions.

A much better approach is to use structured graphics programs: Inkscape,[2] Dia, XFig, gnuplot,[3] and OpenOffice.org[4] are all viable choices for creating diagrams.

Place your graphic files alongside your `.tex` source files. The `Makefile` is already configured to process many common types of graphics files. If these rules do not work for you, or you need assistance, please contact the Proceedings team.

To incorporate graphics in your paper, use the `\includegraphics{}` command. See Figure 3 for an example. This should almost always be part of a "float".

When specifying the name of an image to be included, only the base filename (no path, and no extension) should be given. This allows `latex` will find `.eps` version, while `pdflatex` will use the `.pdf` version.

This year's Proceedings will be in color (or, if you prefer, *colour*). If you choose to use color graphics and figures, please do so effectively, drawing attention to important information and distinctions in your paper. A single box or letter doesn't usually add much information, but does squander resources needlessly.

Once again, please avoid using bit-mapped graphics unless you have no other choice (for instance, a JPG photograph). Structured graphics always produce better printed output.

[2] Saving as PDF seems to work better with Inkscape.

[3] `set terminal postscript eps color` is the command you want to use with gnuplot.

[4] As of 2007, OO.org required that you turn off page numbers, save as a PDF, run `pdfcrop` on the saved output, and then run `pdftops` to get EPS output.

```
% By omitting the extension,
%  - pdflatex finds jwl-page-fig.pdf and jwl-col-fig.pdf
%  - latex finds jwl-page-fig.eps and jwl-col-fig.eps

% Full pagewidth figure, spanning both columns:
\begin{figure*}
\includegraphics[width=0.9\textwidth]{jwl-page-fig}
\caption{The caption appears beneath the figure}
\label{jwl-page-fig-label}
\end{figure*}

% Single-column figure:
\begin{figure}
\includegraphics[width=0.9\columnwidth]{jwl-col-fig}
\caption{Captions appear beneath the figure}
\label{jwl-col-fig-label}
\end{figure}

% Example reference:
See Figure~\ref{jwl-col-fig-label} for single-column use.
Remember that the tilde is a non-breaking space.
```

Figure 3: How to use `includegraphics`

3.2 Trademark and other symbols

LaTeX is able to produce an astonishing number of unusual symbols from the field of mathematics and beyond. You may find it necessary to include the Registered® or Trademark™ symbols in your paper. These can be obtained using the macros `\LegalR` and `\LegalTM`. You may also use the LaTeX2e `\text` constructs: thus, `\textregistered` and `\texttrademark`.

If you need to use asterisks or the Registered or Trademark symbols in a section title—please avoid doing so, it's ugly. If it cannot be avoided, the macros `\titleLegalTM`, `\titleLegalR`, and `\titleStar` are the way to insert the marks and avoid any havoc in the Table of Contents.

3.3 Identifiers and source code

A number of useful macros based on the `url` package are available. These macros cannot be used in floating environments, but they should be used everywhere else—they're easier to type and produce better results than the alternatives. They are:

- `ident` – intended for identifiers, `\ident{some_text}` sets the text in `typewriter` and may break the line at any punctuation. Spaces are deleted.
- `lident` – intended for long identifiers, this works the same as `ident`, but sets the text in a smaller font.
- `code` – intended for short excerpts of code, this works like `ident`, except that spaces are preserved. Lines are not broken on spaces.
- `lcode` – intended for longer excerpts of code, this works like `code`, except that text is set in a smaller font. This probably does not work correctly for multi-line code fragments; consider using the `cprog` package for that.
- `brcode` – intended for excerpts of source code, this works like `code`, except that line breaks may occur at spaces.
- `lbrcode` – intended for excerpts of source code, this works like `brcode`, except that text is set in a smaller font.

Examples are shown in Table 2.

- `\ident{a_long_identifier}` — this example in turn yields `a_long_identifier`
- `\lident|An_Even Lon ger Identifier|` — this in turn yields `An_EvenLongerIdentifier`
- `\lcode{int un_useful(int *a) { return *a; }}` — this yields `int un_useful(int *a) { return *a; }`
- `\lbrcode{int un_useful(int *a) { return *a; }}` — this yields `int un_useful(int *a) { return *a; }`

Table 2: Custom macro examples

3.4 The OLS class

Many thanks go to Zack Weinberg for studying prior years' templates and proceeding to write the `ols.cls` class and other crucial bits of infrastructure for 2005.

The document class for your paper can be one of `galley`, `proof`, or `final`, the latter of which is also the default. Some folks might want to use the `galley` option during the writing process; others might find `galley` to be confusing. In any case, please be sure to submit your paper using the `final` class. The differences are:

- `galley` — All "this doesn't fit" warnings are suppressed, and references appear using the unique identifier rather than the numeric value.
- `proof` — All "this doesn't fit" warnings are active, as are references. Any text that does not fit is marked with an ugly black blob in the margin.
- `final` — All warnings and references are active, and the paper produced will be similar to the one which will be published (but without headers, final page numbers, and such).

The Linux Symposium package also includes several other potentially useful add-ons such as `url`, `zrl`, and `graphicx`. These add-ons are located in the `common` directory, and papers are setup to include them by default.

3.5 Tools

It helps to have the following installed on your system:

- **`tetex`**. The most common TEX package for Linux. Related useful packages include **`dviutils`**, **`xdvi`**, **`dvips`**, and **`ghostscript`**. Be aware that certain distributions package `pdflatex` separately.
- **`transfig`**. Graphics in `.fig` format, useful for figures.
- **`dia`**. Useful for figures. Other commonly used graphics programs include **`transfig`**, **`xfig`**, **`inkscape`**, and **`OpenOffice.org transfig`**. Please note that your export options are crucial, and that we recommend sending along the original, native file as well. You should aim for portability: for instance, certain OpenOffice versions let you choose between EPS with Pango fonts, or EPS — you want the non-Pango EPS for portable printing. As of 2007, the best output from OpenOffice is generated by turning off page numbers, saving as PDF, and running `pdfcrop` on the saved PDF. `pdftops` can then generate EPS from the PDF.
- **`ImageMagick`**. Great for photographs and graphics manipulation & conversion, especially the `convert` program.
- **`xpdf`**, **`evince`**, **`kpdf`**, or **`acroread`** for viewing PDF files. Other viewers can also do a nice job.
- **`pdftk`** if you wish to concatenate PDF pages or perform several other tasks with PDFs. This package is required for building the final Proceedings.
- **`gnuplot`** for drawing graphs based on large quantities of numeric data.

- **dviconcat** is required for building the entire Proceedings, and can be had from the *dviutils* package.

- **emacs** is my editor of choice, in part due to its "LaTeX Fill" mode, which does a very nice job of handling markup and syntax, with automatic line wraps. Please check to see if your editor of choice has a mode for editing LaTeX files; it could save you from tracking down syntax errors and such. (I also use and enjoy `gvim`, especially its *diff* mode.)

Beware any program that cannot export structured graphics, or has troubles respecting standards. These are most commonly found on non-Linux systems; with some of these, you may be better off printing to a hypothetical Apple Laserwriter on `FILE:` and converting the proceeds than in trying to rely on the program doing anything remotely sane and portable.

3.6 Examples

Some examples from previous conferences have been included in this package; hopefully they'll be useful in handling code examples. Reducing everything to `footnotesize` or setting it `verbatim` won't magically make it fit on the page, alas. Have a look in the `example` directory to find these items:

- `bibliography.tex`, `bibliography2.tex`, and `references.tex`. Different ways of citing any relevant works external to your paper.

- `figures.tex`. Different ways of doing figures.

- `includegraphics.tex`. Different ways to include graphics.

- `legalese.tex`. Legal disclaimers. Please note that people want to read technical information, not disclaimers; thus, we set any section of disclaimers in small fonts. To make the Demands Of Lawyers™ easier to deal with, we provide `\LegalTM` and `\LegalR`. The latter looks like This®.

- `multipleAuthors.tex`. Formatting examples for multiple authors.

- `tables.tex`. Different ways to do tables.

4 Submission checklist

To ensure that your paper can be processed with minimum fuss, please be sure to follow these guidelines. Any exceptions should be discussed in advance with the Proceedings team.

1. Avoid bit-mapped graphics, except for photographs or screenshots. See Section 3.1 for alternatives.

2. No proprietary document/graphics formats, please. This especially means MS Office, Visio, or other such tools.

3. Provide original sources. For diagrams, please include the XFig/Dia/Inkscape files. For gnuplot please send the raw data, not the resulting output. If photographs are required, send them at full resolution.

4. Paper size. The papers will be printed on "letter" size paper. Please keep that in mind if you are concerned about page breaks and such.

5. Version control. Please ensure all your source files are placed in the Subversion repository. Commit any changes to the repository as well. If this is not possible, you can send small patches via e-mail.

6. Make no changes to the `common` directory. Do however feel free to send suggestions if you think something can be improved.

7. Use the existing directory structure. Each paper exists in a directory based on the author's last name. The main paper should be called `lastname.tex` and any additional files should be `lastname-file.extension`.

8. Omit file extensions and pathnames in your LaTeX source, please. By omitting the path and just saying `\input{lockhart-abstract}`, a paper can be built from both its directory and from its parent directory. For graphics, omitting the extension lets `latex` or `pdflatex` pick its preferred input format for the best possible results.

9. Do ***not*** use sans-serif fonts, or go changing global font sizes. We're using Times Roman for body text, in a consistent size that's comfortable for hardcopy. Likewise, please don't go haywire with italics. I once received a huge collection of tables, each of

which set the font size and face on an item-by-item basis. *Incorrectly*.

10. Those of you who like to begin lines of code with commas: we are typesetting the code with the comma attached to the preceding identifier, as most publishers do. Feel free to post your preferred version to the web and to refer to it in the paper.

11. If possible, please avoid trivial new macros. Should you need to add something, though, please use `\providecommand` rather than `\newcommand`, and try for a relatively unique name (papers tend to blur together during long editing sessions).

12. Trivia note: generally speaking, it takes longer to edit a submission from a TEXspert than plain, unmarked ASCII. If you consider yourself a LATEX expert and love to write fancy new commands, please consider contributing clean-ups or well-tested new features for the infrastructure rather than customizing the daylights out of your submission. Thanks!

4.1 Language nitpicks

The following language issues come up frequently. Please review your paper for these issues prior to submission.

1. Proofreading is a wonderful thing, and every bit of it you (or any guinea pigs you can persuade) do is a Good Thing.

2. Be happy, know your homonyms. There, they're, their. To, two, too. Your, you're. And so forth. Spelling checkers show their limitations on this...

3. In particular, remember that *it's* is an abbreviation for *it is*, while *its* is possessive.

4. When listing items in a series, you should write: *a*, *b***,** and *c*. Never *a*, *b* and *c*. This rule makes it much simpler when you must use complex values of (for example) *b*. For truly long constructs, you may use a semicolon as a delimiter rather than a comma. This is especially useful if item *b* itself contains commas.

5. Some phrases should be hyphenated—for instance, when you're using an adjective to modify another adjective, or a noun that appears before another. A high-performance system; a win-win situation; a high-level loop transformation; a slow-moving train, but a slowly moving car; that sort of thing. Most of the time, people will still be able to parse the results easily even if the sentence isn't perfect.

And remember, it's only typesetting, not rocket science. Or hacking compilers or kernels. `:-)` Have some fun along the way...