Travels in \TeX\ Land: A Macro, Three Software Packages, and the Trouble with \TeX

David Walden

Abstract

In this column in each issue I muse on my wanderings around the \TeX\ world. In this issue, I deal with three unrelated topics: I describe how a small macro works that I decided to try to understand, I briefly describe my experiments with three \TeX\-related software packages, and I give my perspective on why lots of people find \TeX\ difficult.

1 Understanding a small macro

When I need some feature of \LaTeX\ or \TeX\ that I don’t already know, I usually just find an example of what I want to do and copy it without understanding it. If I need to change it, I do it hit and miss, still mostly without understanding. However, occasionally I decide to try to understand an example I have found. In these cases I like to write up what I’ve learned, to help me be sure I’ve really got it, and, who knows, it may help out someone else, too.\footnote{This is a pretty fragmented way to learn stuff, but it is mostly the way I do it.}

For years I heard the words “kern” or “kerning” without knowing what they meant. More recently I understood it meant something having to do with the tightness of spacing of letters in a word. Recently I saw a simple macro — a definition for typesetting the \LaTeX\ logo that involves kerning — and decided that it was time to actually understand kerning. This example had such an extreme example of kerning that I figured the example would make clear how kerning works, and perhaps I’d also learn some other things. Here’s the example:
\def\LaTeX{%
L\kern-.36em
{\setbox0=\hbox{T}\% \vbox to \ht0{\hbox{\the\scriptfont0 A}\vss}\}% \kern-.15em
\TeX
}

Of course, the above example is the simplest kind of \TeX\ macro definition. The \def\ command says that the control sequence \LaTeX goes to be replaced with the text between the open brace at the end of the \def\ line and the matching close brace six lines later to produce the iconic version of La\TeX, i.e., \LaTeX. The first letter of the replacement text is the letter L, which is obviously a good start. The next replacement text is \kern-.36em. To see what the \kern\ command did, I defined the macro
\def\La{L\kern-1em% 
 a
}
Calling this macro resulted in
aL
In other words, \kern followed by the value -1em tightens the spacing between the letter L and the letter a so much that the a is moved all the way to the left side of the L. The em units are traditionally about the width of a capital letter M in the current font.\footnote{Paul Abrahams et al., \TeX for the Impatient, Addison Wesley, 1990, page 60 (also available free on the web from http://www.tug.org/ftp/tex/impatient/book.pdf). However, the em units, in fact, are specified as part of every font and even can be changed by the user with the \fontdimen\ command.} Negative values of \kern tighten the spacing between letters and positive values widen the spacing.\footnote{This is how \kern works in \TeX's horizontal mode; in vertical mode does something different.} No doubt someone did a lot of trial and error with values for \kern to choose the value that moves the letter A to have just the right amount of overlap with the letter L.

Looking at how \LaTeX prints, it is clear that the next part of the \LaTeX macro, i.e.,
in some way moves the letter A up and makes it smaller. Here’s how that works.

- A box in \TeX{} is a two-dimensional shape having a height above the Baseline, a depth below the Baseline, and a width from a Reference point.\footnote{According to chapter 11 of Donald Knuth’s \textit{The \TeX{}book}.} \TeX{} makes pages by “gluing” boxes containing individual characters together into bigger boxes that in turn are glued together into bigger boxes until a page is filled. The bigger boxes can be hboxes (for horizontal sequences of characters) or vboxes for vertical stacks of hboxes.\footnote{To be more precise, hboxes and vboxes can be nested arbitrarily.}

- \TeX{} has 256 box registers in which boxes can be saved.\footnote{Ibid, page 120.} Thus, the construction \texttt{\setbox0=\hbox{T}} sets the value of box0 to be a horizontal box with the single letter T in it.

- \texttt{ht0} has the value of the height of box 0, i.e., the height of a letter T.\footnote{Ibid, page 120.}

- The construction \texttt{\vbox to \ht0{\ldots}} creates a vbox whose height is the value the height of box 0 and whose content is an hbox containing the stuff within the braces.\footnote{Ibid, page 77.}

- \texttt{\scriptfont0} is an identifier for script type font 0. Juxtaposing \texttt{\the} in front of \texttt{\scriptfont0} selects the specified font, i.e., it is as if the control sequence for the specified font had been written. So, \texttt{\{\the\scriptfont0 A\}} says that the letter A should be in the format of scriptfont0.\footnote{Ibid, page 80.}

- The \texttt{\vss} command that is also within the vbox adds enough vertical space to the vertical box to fill it up exactly,\footnote{Ibid, page 153.} since \TeX{} doesn’t like boxes to have

\begin{verbatim}
\setbox0=\hbox{T}%
\vbox to \ht0{\hbox{\the\scriptfont0 A}\vss}}%\end{verbatim}
too little or too much “stuff” in them.\footnote{An experiment of leaving out \texttt{vss} showed no visible change in the resulting printout; but Karl Berry told me that, in this case, its purpose is to avoid an underfull or overfull box message in case the height of $A$ is different than the height of $T$.}

\begin{itemize}
\item Thus, that letter $A$ in that font, which apparently is a little font, is placed by the \texttt{vbox} at the height of a letter $T$ as we see is in the \texttt{LAT} part of \texttt{LATEX}.
\end{itemize}

Returning to the whole macro

\begin{verbatim}
\def\LaTeX{%
  \L\kern-.36em
  {\setbox0=\hbox{T}%
    \vbox to \ht0{\hbox{\the\scriptfont0 A}\vss}}%}
  \kern-.15em
  \TeX
}
\end{verbatim}

we see it ends by decreasing the tightness of the spacing using another \texttt{kern} command, and then the \texttt{\TeX} macro\footnote{\textit{The \TeX}book, page 66.} is called producing \texttt{\TeX} to complete \texttt{LATEX}.

2 Three software packages

I recently tried three new (to me) \texttt{\TeX}-related software packages, installing each on my Windows 98 system.

Pro\texttt{\TeX}xt

I hadn’t upgraded my MiK\texttt{\TeX}-WinEdt-Ghostscript-Adobe-Acrobat configuration in at least seven years. So it seemed about time in mid-May to try to get things up-to-date. Until I read the last issue of \textit{TPJ}, I had always wondered what to do with the CDs and DVDs that come yearly from TUG. However, in the last issue, there was an “Ask Nelly” answer from Karl Berry explaining, among other things, what the pro\texttt{\TeX}xt CD is about. So, I went to Karl’s answer again and there found a link to the pro\texttt{\TeX}xt home page on the TUG web site. That told me to start the pro\texttt{\TeX}xt CD
and follow the PDF-based instructions on the CD. I got the instructions, printed
them out, and read them (as they recommend, although it is not my usual practice
to read directions).

The next morning, I again loaded the pro\TeX\t CD and began to follow the di-
rections. The directions consist of blocks of explanatory text and instructions and
places to click to move to the next step in the installation process. First the direc-
tions said to delete my old MiK\TeX, WinEdt, Ghostscript, and Ghostview installa-
tions. I did this although it was pretty scary—I would have nothing to go back to
if the installation of the new stuff failed.

Next, I installed MiK\TeX to my hard drive but I ignored its default “Large”
option and instead selected the “All” option, which took 2.5 hours to load and
install. I was pleased that it allowed me to put the root $\text{\texttt{\textbackslash texmf}}$ hierarchy on my
C drive where I keep applications and to put my $\text{\texttt{\textbackslash localtexmf}}$ hierarchy on my E
drive where I keep application files. The instructions also tell you how to change
some MiK\TeX configuration files to handle 8 1/2 by 11 paper, and that was no
problem. They didn’t mention that I needed to set up my printer offsets for $\text{\texttt{dvips}}$
which I figured out by bashing around in the $\text{\texttt{dvips}}$ configuration file.\footnote{I found the following parameter line in the $\text{\texttt{dvips}}$ configuration file by searching for the word
“offset”: 0 0pt,50pt. The capital letter 0 apparently stands for “offset,” and I printed out lots of
trial sheets of paper as I tried many values in the rest of the command line before settling on zero
and fifty as good enough.}

The installation process allows you to install $\text{\texttt{\textbackslash\TeXnicCenter}}$ or WinEdt and three
of its plugins. Since I already used WinEdt, I chose it and clicked to install it. These
days WinEdt is supposed to automatically find and link to MiK\TeX, Ghostscript,
etc. Its installation also was supposed to preserve my WinEdt registration code. It
did find the other applications, but it did not find my old registration code (per-
haps my deletion of my old WinEdt files had been a little too brute force); fortu-
nately, I still had a copy of the email message with my WinEdt registration, but I
had to remember how to give it to WinEdt since this is not assumed by the pro\TeX\t
instructions. After a day of use, I also had to bash around in WinEdt and its con-
figuration files for an hour or more trying to figure out how to tell WinEdt to use
the latest Adobe Reader on my machine when I click the Adobe Reader icon rather
than using my Acrobat 3.0 version of Exchange to display PDF files.\footnote{There are no general lessons to be learned by me describing the completely ad hoc path I
stumbled along until I managed to point WinEdt at the latest Adobe Reader. It is better to hope
that WinEdt’s automatic configuration capability does the right thing.}
All in all, the installation went relatively smoothly, did not take too many hours (compared with various worse case scenarios I imagined), and I again have a working, much more up-to-date environment for doing my \TeX work. I do think the pro\TeXt instructions could address a few more of the possible eventualities (but maybe it is better to keep them simple). If I was a brand new \TeX or WinEdt user, it would have probably taken me a lot longer to figure out how to make the several detailed changes I had to make that were not covered in the pro\TeXt instructions.

**Win\TeX**

When someone suggested doing a 30-day trial of Win\TeX (http://www.tex-tools.de/), I jumped at the chance. Although I didn’t intend a systematic or deep evaluation, I was interested in it, based on the developer’s web site description of it as a “\TeX / \LaTeX editor with MS Office look and feel.”

The download and installation worked just fine, and I used Win\TeX on a few little things for a week or so. It all worked quite smoothly with lots of nice \LaTeX structure and features visible to minimize the amount of \LaTeX keyboarding needed (and keyboarding mistakes). Win\TeX did seem a little slow to launch compared with my usual text editor. I didn’t dig deeply enough to understand how powerful the Win\TeX editor is (e.g., whether is has regular expression searches). All in all and given the minimal trial I did with it, it looks like a plausible choice for people using \LaTeX.

I do have one gripe. Win\TeX captures the meaning of one (or more?) file extensions, .tex in particular. Now this is not something I can specifically hold against Win\TeX — most Windows applications seem to do this. And while I was testing Win\TeX, it was not so bad that when I clicked on a .tex file, Win\TeX launched. However, after I uninstalled Win\TeX, clicking on a .tex file gave an error message saying the application couldn’t be found. It is not too hard to reassign another program to the .tex extension if you know how, but for me it was a struggle to figure out how again. I wish Windows application trial installations would not capture any extension that might be used by another application and would leave doing this until an explicit user command has been given.
Word2TEX

I have a project where I need to convert twelve lengthy papers that their authors wrote in MS Word into \LaTeX{} for inclusion in a collection of papers for which I am using \LaTeX{} to typeset the volume. Thus, I looked around (again) for a program to automatically convert Word to \LaTeX{} and found Word2TEX.\footnote{For more Word-to-\TeX{} conversion options, see \url{http://www.tug.org/utilities/texconv/}.} The Word2TEX web site (\url{http://www.chikrii.com}) has some decent testimonials for Word2TEX; and I also asked \TPJ{} editor Lance Carnes about it since his company (\url{www.pctex.com}) is a reseller of Word2TEX, and he told me he has heard good things from his customers. So, I downloaded and installed the 30-day trial of Word2TEX, which severely limits the number of equations, figures, and tables that will be converted from one Word file. The installation went smoothly.

All one does to convert a document from Word to \LaTeX{} is open the .doc file with Word and then save it specifying the .tex file type, and a \LaTeX{} file is saved. My test case was a 25-page document with no math, three figures, and no tables, so the artificial limits of the trial version of the program did not get in the way much. The conversion was quite sensible, although the generated \LaTeX{} was a little wordy. The resulting file was well-formed \LaTeX{} that compiled immediately from \LaTeX{} to a .dvi file that I could view with my previewer. I did have to make some changes to the resulting \LaTeX{} to change what I wanted the output to look like that was different than the Word file looked.

I don’t know what would have happened if I had pressed Word2TEX to do harder tasks. It did seem sufficient for my task, and I gave them my credit card number and ordered a copy that would not have the artificial limitations the trial version had.

3 The trouble with \TeX{}

For the last issue of \TPJ{}, Arthur Ogawa wrote a long opinion piece on some of the problems with the \TeX{} interfaces (“In my opinion: \TeX{}'s Interface Challenges, \TPJ{} Journal, issue 2, 2005). Arthur presented a sophisticated analysis of what could be done to make \TeX{} easier to learn and use. I have a more primitive take on why new users of \TeX{}, \LaTeX{}, and the rest find \TeX{} hard to learn.
First, \TeX{} has become very fragmented with lots of distributions, styles, packages, platforms, etc.\footnote{Such fragmentation typically happens in every field after a successful innovation. New companies are started with different or better versions of the initial innovation, many after-market options are available, etc.} Of course, such capabilities are also part of \TeX{}’s power.\footnote{I might argue that it is \TeX{}’s astonishing interoperability of various components, releases, formats, etc., that allows diverse distributions (the fragmentation I am referring to) from different vendors and providers that \TeX{} users can move between rather than there just being different products that may not interoperate so well as is the case with normal product fragmentation. In other words, all that good stuff provides part of the complication we see ourselves faced with.} While there are on-going efforts to do some consolidation (such as creating more standard ways of using CTAN), I see it as inevitable that there will be to-some-extent competing versions of \TeX{} available and a vast array of sometimes conflicting add-ons.

Second, \TeX{} looks “different” to typical word processor users because WYSIWYG and Word have won the word processing game (at least for now).\footnote{This also typically happens after a successful innovation and its following fragmentation — next comes consolidation around one or two offerings, and everyone else eventually goes out of business or hangs on with minor market niches.} \TeX{} didn’t win the typesetting game either,\footnote{Which, apparently, has been substantially lost to systems like Quark.} probably because it doesn’t work in the way that professional typesetters are used to — doesn’t let them fuss with the sorts of things they like to fuss with. Despite many efforts to make \TeX{} more graphical to use (e.g., Scientific Word, BaKoMa \TeX{} Word, WinTeX, LyX, the WinEdt “table designer,” etc.), I think much \TeX{} use will remain command driven with distinct edit and compile steps.

Third, most people never want to learn more than the minimum about anything they have to use or do. We learn to use a couple of buttons on our microwaves but no more. We go to one supermarket and not others because we know where stuff is in the first supermarket (and we are annoyed when the supermarket rearranges the aisles). We don’t know how to use most of the dials and controls on the dashboards of our cars. Etc.

WYSIWYG and Word (i.e., the point-and-click GUI interface) have won in the word processing world partly because one can use it like a typewriter without knowing anything other than how to save a file with a new name. Most people I
know, whether they are PhDs in science or home users who know next to nothing about computers, use Word like a typewriter. They tab and space to center a line, and they tab to do indents including nested indents; they have no knowledge of the ruler-bar for setting margins and first line indents. They do footnotes without knowing about automatic numbering or even how to have them automatically appear at the bottom of the same page as the footnote reference. They find and edit by using the arrow, delete, and backspace keys; they can use Cut and Paste on the menu but don’t know anything as simple as control-X to cut and control-V to paste. And so on. People use Word because it is the standard, and they don’t have to learn anything and prefer their primitive known ways of doing things to learning anything new.

To reiterate, people generally don’t want to learn anything new. Of course, with \TeX you can’t know so little.

The key problems, as I see them, to \TeX having any chance of gaining new popularity are that \TeX requires massive consolidation, simplification, and a return to being perceived as the state of the art for word processing and typesetting. Unfortunately, this would require a development effort that is too big to have much chance of getting started and less chance of being completed successfully.

Yet, there are lots of users who chose to use \TeX and stick with it for some reason: (a) we need its power, (b) a thesis supervisor insisted, (c) we can’t abide Microsoft or using Word, (d) using or supporting \TeX somehow provides us a living, and, (e) a few may just love intricacy or \TeX’s elegance.

My current thought is that we should stop worrying about why \TeX isn’t more popular and simply concentrate on making its use the maximally pleasant and productive experience we can for those people who, for whatever reason, choose to use \TeX.

---

21 People I know who were users of Nota Bene and Word Perfect stay with them relentlessly because they know them and don’t want to learn Word. Personally, I upgrade Word as seldom as possible because I don’t want to learn how Microsoft has changed it. I know people who stayed with WordStar for years after it was dying.
Acknowledgements

Karl Berry read a draft and informed me regarding some nuances. The comments of the anonymous reviewers were very helpful and reassuring. Tristan Miller’s and Will Robertson’s comments were also useful. David Elliott caught a couple of typos at the last minute.