Warnings

Initiation rites

Barbara Beeton

Initial conditions always have to be looked at carefully if one wants to avoid nasty surprises, a fact known all too well to students of fluid dynamics, programmers, and other assorted technoids. Boundaries between TEX mode changes and other TEX structures are no exceptions to this rule.

Paragraph beginnings

Automatic hanging indent. There are times when one wants several successive paragraphs to be hanging indented. It's a great nuisance to have to type the instructions for a hanging indent into every paragraph, and though it's easy enough to make a simple substitution macro

```
\def\X{\par\noindent
    \hangindent\parindent\hangafter1 }
```

using it at the beginning of every paragraph is also a nuisance.

But wait — there's `\everypar`. Why can't we use that to make the hanging indented style automatic? Most of the time we can, but there are occasional problems, as the following example shows.

```
%% for a test, make it small and loose

\parindent=0sp \hsize=2.5in
\hfuzz=2pc \parskip=.5pc

\everypar{\hangindent.5in \hangafter=1 }

{\it H}owdy boys and girls.
  How are you doing today?

% But, wow!  A strut fixes it!
\strut {\it H}owdy boys and girls.
  How are you doing today?

% Also, not nesting it works
\it H\rm owdy boys and girls.
  How are you doing today?

% Ordinary paragraphs work fine too.
Well, now is the time for all good men
to come to the aid of their party.
```

Here's what the output looks like.

```
Howdy boys and girls.  How are you
doing today?

Howdy boys and girls.  How are you
       doing today?

Howdy boys and girls.  How are you
       doing today?

Well, now is the time for all good
       men to come to the aid of
       their party.
```

Not exactly what we had in mind.

Why has the hanging indent disappeared from the first paragraph? It turns out that TEX applies `\everypar` immediately after it encounters a token in vertical mode that will cause it to shift into horizontal mode (*The TEXbook*, p. 105). In this example, the initial "H" is such a token. Unfortunately, this is inside a group, and all the default conditions are restored upon exiting the group; the transcript of a trace (slightly edited) shows this very clearly:

```
{begin-group character {}

\it ->\fam \itfam \tenit
{\fam}
{select font cmti10}
{the letter H}
\everypar->\hangindent .5in \hangafter =1
{horizontal mode: \hangindent}
{\hangafter}
{the letter H}
{end-group character }}          ***
{restoring \hangafter=1}
{restoring \hangindent=0.0pt}
{restoring current font=\tenrm}
{restoring \fam=0}
{the letter o}
```

Here are some more things that won't work, because they don't put TEX into horizontal mode:

- `\relax`;
- an empty `\hbox` or `\vbox`; this has an additional nasty side effect — it leaves a "blank line" above the paragraph that is occupied by nothing but the empty box;
- an `\hrule` with `width0pt`, with an effect similar to that of an empty box; in this case, extra blank space depends on the height of the `\hrule`.

And here are several more things that *do* work:

- `\leavevmode` (the most neutral option);
- `\noindent`;
- `\indent` (because `\parindent=0sp`);

- a `\vrule` with `width0pt`;
- math mode with no contents: `$ $`; just be sure not to leave any space between the closing dollar sign and the first word of the sentence, or it will be there in the output.

Although this is a warning about beginnings, it seems prudent to put in just a few words about endings, at least of hanging indented paragraphs.

A paragraph doesn't actually end until TeX comes to `\par`, `\endgraf`, or some implicit shift to vertical mode, e.g. `\vskip`. If a paragraph that's supposed to be hanging indented is in a group, and the group ends before the shift into vertical mode, the hanging indentation is lost. TeX only applies certain "shape" adjustments (among them `\leftskip`, `\rightskip`, `\baselineskip`, and the like) upon the transition to vertical mode, so whatever those settings are at that transition, those are the ones that take effect.

Paragraphs that begin with boxes. This has already been touched on above. Neither an `\hbox` nor a `\vbox` will cause TeX to enter horizontal mode. (If TeX is already in horizontal mode, it will stay that way.) Instead, the box is set on a line by itself, at the absolute left margin. If you are using `\llap` to tack something on to the left of the first word in a paragraph, you should remember to precede it by `\leavevmode`; the answer to exercise 14.28 in *The TeXbook* works because the first command activated by `\strut` is `\unhcopy`, which causes TeX to go into horizontal mode, and the answer to exercise 14.29 works because `\everypar` is not invoked until TeX is already in horizontal mode.

Table cells

The start of data in a cell of a table can also be an invitation for something to go wrong. The reason is simple: TeX will always expand the first token to see if it is `\noalign` or `\omit`, to make sure that items that must be processed before the template, or that ignore it, are handled properly. *The TeXbook* (p. 240) tells us that macros are expanded until the next non-space token is found; if it is not `\noalign` or `\omit`, it will be put back to be read again after the first part of the template has been read. However, the order of expansion will now be something other than intended, and for some macros whose proper application depends on their being expanded in a particular order, the results may be unexpected.

An example from the *AMSFonts User's Guide* will illustrate the problem. I wanted to present the cyrillic transliteration conventions in the form of a table. Since two columns (upper- and lowercase) would always be cyrillic, it seemed obvious that the cyrillic command should be put into the template. Here's what I tried first.

```
\halign{{\cyr#}\hfil\quad&#\hfil...\cr
      \u\i & \u\i ...\cr
      ... }
```

What I wanted was

  й ĭ

What I got was

  Ю ĭ

This only makes sense if you happen to know the layout of the cyrillic font, and even then not much sense at first.

Here's what happened. `\u` is expanded and determined to be an accent. Unfortunately, `\cyr` in the template hasn't been expanded yet, and one of the functions of `\cyr` is to invoke some special macros that treat certain combinations of accents and letters as single letters in the cyrillic alphabet. So the accent is assigned as a separate character (to be found in the position `\char'25` as in the default text font), and it is superimposed according to the usual accenting rules over whatever is found as `\char'20`.

Actually, this is fixed relatively easily; all that is needed is to begin each cyrillic cell with `\relax` (or some other "harmless" control sequence, as Don Knuth expressed it when I asked if he'd please explain to me what was happening). So:

```
\relax \u\i & \u\i \cr
```

As it turned out, I finally used another technique entirely to create the table. But I learned the dangers of macros being expanded out of order, and was also prepared for the questions that arose when users of the cyrillic font hadn't included the special accent macros in their definition of `\cyr` — the output looks just the same! Dreadful!

Moral: If the contents of some cells in your table don't look quite like what you expected, it may be worth putting `\relax` at the beginning and trying again before you start digging into the macros.