Warnings & Limitations

Another Hangup

In the last issue, you were warned that repetition of a \let statement can cause \TeX{} to hang. It has been pointed out that \ifx can be used to detect recursion (provided you are using a recent enough version of \TeX{}; see the errata list, extensions since June 30, 1981).

Another way to make \TeX{} hang is to set to a negative value one of \TeX{}'s integer parameters that is expecting a positive value. For example, \char3 and \char13 (which will become \dpen, \dp and \adjpen in \TeX{}82) have caused the SAIL version of \TeX{} to loop at the Math Society (we are still running with a version of March 1981). This will probably be trapped in \TeX{}82, but I didn't see it described in the differences list.

Barbara Beeton

MACRO

MACRO INDEX

The following list catalogues macros that have appeared in TUGboat. Entries are listed by volume, number, and page as well as author's name. Items that could not be categorized by an obvious headword have been listed under "miscellaneous". Many items refer to parts of large macro packages; users of other packages may find them valuable models for macros of their own.

Readers' comments on the format as well as the contents of this index are welcome.

ACM style    II.1 61, 62-63 A. Keller
Addresses     II.1 54 B. Boiten
              II.2 A-35 M. Diaz
Appendices    II.2 A-21 M. Diaz
Baseline, set to top of box II.1 60, 77 A. Keller
Bibliography  II.2 A-25 M. Diaz
Boxes         II.1 59, 73 A. Keller
Box numbers, automatic allocation III.1 33 M. Plass
Branching, see if
Capital letters
large — at beginning of paragraph II.1 60, 78 A. Keller

Roman numerals
II.1 120-121 P. Milligan, L. Price

Centering a sequence of lines II.2 A-13 M. Diaz
This is probably the best place to point out what I really wanted was for the height above the highest character to equal the depth below the baseline. As it is, there is 2pt above the highest character and 2pt below the deepest. I probably wouldn't have given up except that cmr30 was already pretty tight on the page and page breaking was not appetizing. Actually when it was working this well I was pretty relieved.

leol labels the top, again in cms10. chw, colw and setw are used to find the maximum width of any character in the font. getw takes the maximum over the set of characters in the font, the width of the 0 used in labeling the columns, and 1em in the font (maybe unnecessary) then sets the variable unit to $1\frac{3}{8}$ this value. The 1vu is used as the width of each character cell.

Finally, table is defined to use the given character to define the font, set the font and build the table. The hbox has glue to center if possible but to left justify with right overfilling forgiven if necessary. The font name is included in cms10. The top label and the top rule for the font cell set are followed by the sixteen cellrows.

Editor's note: The two tables which follow were pasted up from Varian copy generated at the Math Society. A few changes were necessary: new letter codes were assigned to the two fonts because of conflicts with codes already assigned to preloaded fonts; cmr28 does not exist at the Society, so cmr30 was substituted.

We discovered after looking at the first output that this routine neatly illuminates probable errors in a couple of METAFONT descriptions. In the cmr30 table, row '000 has too much depth, and character '121, "Q", has no depth where one would have expected it. On checking the METAFONT descriptions, we found that the depth of the "Q" has disappeared (presumably accidentally—it was present in the original published description of the Computer Modern family), and that character '002, "Θ", has always been assigned a depth equal to that of a comma.