How to Avoid Writing Long Records to TeX's \write Streams

Peter Breitenlohner

In his article 'Macros for Indexing and Table-of-Contents Preparation' in *TUGboat* 10, no. 3 (1989), pp. 394–400, David Salomon mentions a problem with very long records written to one of TeX's `\write` streams (p. 399): "... Since each line of the table of contents goes on the file as a record, its size is limited and, as a result we cannot have chapter or section names which are too long. ..."

There seems to be a similar problem in LaTeX, since LaTeX users frequently request that TeX implementations be able to write very long records (for CMS up to 1024 characters).

There is, in fact, a very simple method to avoid this problem: split the output into several records at all those points where a new line starts in the user's input. The macro `\chap` defined below (to be used with `plain.tex`) demonstrates how this can be done. The argument of `\chap` is typeset in bold face (with line end characters converted to spaces) and written (`\immediate`ly) to a file.

```
\newwrite\ind
\def\chap{\bgroup
  \catcode`\^^M\active \chapx}
\begingroup
  \catcode`\^^M\active
  \gdef\chapx#1{%
    \let^^M=\relax%
    \newlinechar`\^^M%
    \immediate\write\ind{%
      \string\ch:#1\string\\}%
    \let^^M=\space%
    \par{\bf#1}\par\egroup}
\endgroup

\immediate\openout\ind=\jobname.ind
\chap{This is
   one (immediate)
   potentially
   very long text}
\immediate\closeout\ind \vfill\break
```

The definition of a similar macro `\Chap`:

```
\newwrite\inx
\def\Chap{\bgroup
  \catcode`\^^M\active \Chapx}
\begingroup
  \catcode`\^^M=\active
  \gdef\Chapx#1{\bgroup%
    \lccode`\*`\^^M%
    \lowercase{\egroup \def^^M{*}}%
    \edef\x{\string\ch:#1\string\page}%
    \write\inx\expandafter{\x%
      \number\pageno\string\\}%
    \let^^M=\space%
    \par{\bf#1}\par\egroup}
\endgroup
\newlinechar=`\^^M

\openout\inx=\jobname.inx
\Chap{This is
   another (delayed)
   potentially
   very long text}
\closeout\inx \vfill\break
```

demonstrates that things are only slightly more complicated for a delayed `\write`.

◇ Peter Breitenlohner
Max-Planck-Institut für Physik
München
Bitnet: PEB@DMOMPI11

Tutorials

Forward References and the Ultimate Dirty Trick

Lincoln K. Durst

In this tutorial, we pick up where we left off in the first episode: See *TUGboat* 10, no. 3 (November 1989), pages 390–394.

The last page, 401, of Appendix D ("Dirty tricks") of *The TeXbook* describes what surely deserves to be called the ultimate dirty trick. Under the heading "*Syntax checking*" the creator — of TeX, of course! — tells us how to disable the output routine, abolish all fonts, ignore all line and page breaks, and otherwise have TeX race through a file doing nothing more than executing the macros it encounters and can recognize. The original intention for providing this feature, as the name makes clear, was to locate typographical or other errors in the names of macros: Any macro