

Macros

The bag of tricks

Victor Eijkhout

The plain \TeX `\loop` macro has been a headache for as long as it has existed. Already in *TUGboat* #2 of 1987, Alois Kabelschacht gave an improved implementation of this macro, and there are regular questions about it on the \TeX newsgroup. The main problem is that the original implementation, which is used as

```
\loop ... \if ... \repeat
```

suggests that

```
\loop ... \if ... \else \repeat
```

should also be possible, which it is not.

The problem lies in the implementation

```
\def\loop#1\repeat
  {\def\body{#1}\iterate}
\def\iterate{\body \let\next\iterate
  \else \let\next\relax \fi \next}
```

which already contains an `\else`, so there can not be another one at the end of the body.

A simple solution is

```
\def\iterate
  {\let\next\relax \body
  \let\next\iterate \fi \next}
```

However, this presumes that in the body the control sequence `\next` does not get redefined; it would be better to use a unique name such as `\nextloop`.

The suggestion in that old *TUGboat* issue is

```
\def\loop#1\repeat{%
  \def\iterate{%
    #1\expandafter\iterate\fi}%
  \iterate \let\iterate\relax}
```

which can contain `\else`.

Another solution comes from David Kastrup, who wrote an `\ifnot` macro for inverting the conditional:

```
\def\ifnot#1{#1\else
  \expandafter\expandafter\fi
  \iffalse\iftrue\fi}
```

which can be used as, for instance,

```
\loop ...
  \ifnot{\ifeof\stream} ... \repeat
```

This macro is worth studying for a moment: let us see what happens to

```
\ifnot\iftrue / \ifnot\iffalse
```

First of all the true case: the expansion is

```
\iftrue\else\expandafter\expandafter\fi
  \iffalse\iftrue\fi
```

and everything from `\else` to the first `\fi` gets skipped; what's left is

```
\iffalse\iftrue\fi
```

which is basically `\iffalse`, that is, the negation of the original `\iftrue`.

In the false case,

```
\iffalse\else
  \expandafter\expandafter\fi
  \iffalse\iftrue\fi
```

becomes

```
\expandafter\expandafter\fi
  \iffalse\iftrue\fi
```

The first `\expandafter` reaches to the `\fi`, which is taken to conclude the original `\iffalse` conditional. We are left with

```
\expandafter\iffalse\iftrue\fi
```

Here `\expandafter` eliminates the `\iftrue`, and \TeX notes that an `\iftrue` conditional has started. Next,

```
\iffalse\fi
```

expands to nothing, and the net result is that we are now in an `\iftrue` conditional, the negation of the original `\iffalse`.

A pretty impressive macro which can of course also be used outside the context of `\loop`. In closing let me remark that I recently wrote a drastic revision of the `\loop` macro, which you can find as `repeat.tex` in the usual archives.

◇ Victor Eijkhout
 Computer Science Department
 111 Ayres Hall
 University of Tennessee
 Knoxville, TN 37996-1301 U.S.A.
victor@eijkhout.net