

Loopy.TeX

J. E. Pittman

Recently, I encountered an application that required a set of nested loops and local-only assignments and definitions. T_EX's `\loop... \repeat` construction proved to be inadequate because of the requirement that the inner loop be grouped. To solve the problem, I wrote a general purpose integer 'for loop' macro, the syntax of which is simply:

```
\forcount\cname = start to
    finish by increment do
    body of the loop
\endfor\cname
```

The *cname* given above must be defined as a count register by a `\countdef`, `\newcount`, or `\declarecount` macro.

```
\def\forcount #1{\relax
\def
    \for #1=##1to ##2by ##3do
        ##4%
        \endfor #1%
{\relax
#1=##1\relax
\ifnum ##3>0
    \whilenot #1\ifnum ##2<#1do
        ##4%
        \advance #1 by ##3\relax
        \endwhilenot #1%
    \else
        \while #1\ifnum ##2<#1do
            ##4%
            \advance #1 by ##3\relax
            \endwhile #1%
        \fi
    }%
\for #1%
}%
%
\let\endwhilenot=\fi
%
\def\whilenot #1{\relax
\def
    \whilenotloop#1 ##1do
        ##2%
        \endwhilenot #1%
{\relax
\expandafter\def\cname whilenotbody\string#1\endcsname{##2}%
\expandafter\def\cname whilenotloop\string#1\endcsname
{\relax
##1%
    \let\next=\relax
\else
```

The 'for loop' macro utilizes general-purpose while and while-not loop macros, the syntax of both is:

```
\while\cname conditional do
    body of the loop
\endwhile\cname
```

The *cname* can be any control sequence name that is locally unique.

A listing of the file `loopy.tex` is given in figure 1. An example file which generates a simple multiplication table and its output are shown in figures 2 and 3.

The definitions of a set of 'declare' macros, which function like non-global 'new' macros, is given in figure 4.

```

\csname whilenotbody\string#1\endcsname
\expandafter\let\expandafter\next
\csname whilenotloop\string#1\endcsname
\fi
\next
}%
\csname whilenotloop\string#1\endcsname
}%
\whilenotloop#1
}%
%
\let\endwhile=\fi
%
\def\while #1{\relax
\def
\whileloop#1 ##1do
##2%
\endwhile #1%
{\relax
\expandafter\def\csname whilebody\string#1\endcsname{##2}%
\expandafter\def\csname whileloop\string#1\endcsname
{\relax
##1%
\csname whilebody\string#1\endcsname
\expandafter\let\expandafter\next
\csname whileloop\string#1\endcsname
\else
\let\next=\relax
\fi
\next
}%
\csname whileloop\string#1\endcsname
}%
\whileloop#1
}

```

Figure 1. Listing of the macros for looping.

```

\beginboxes{}
\declarecount\x
\declarecount\y
\declarecount\z
\column{\leftrulewidth=1.2pt \rightrulewidth=1.2pt}
\forcount\x = 1 to 11 by 1 do
\column{\leftrulewidth=0pt \rightrulewidth=0.4pt}
\endfor\x
\column{\leftrulewidth=0pt \rightrulewidth=1.2pt}
\row{\toprulewidth=1.2pt \bottomrulewidth=1.2pt}
\entry{${\times}$}
\forcount\x = 1 to 12 by 1 do
\entry{\number\x}
\endfor\x
\forcount\y = 1 to 12 by 1 do
\ifnum\y=12

```

```

        \row{\toprulewidth=0pt \bottomrulewidth=1.2pt}
    \else
        \row{\toprulewidth=0pt \bottomrulewidth=0.4pt}
    \fi
    \entry{\number\y}
    \forcount\x = 1 to 12 by 1 do
        \z=\x
        \multiply\z by \y
        \entry{\number\z}
    \endfor\x
    \endfor\y
\endboxes

```

Figure 2. Listing of a loopy example.

×	1	2	3	4	5	6	7	8	9	10	11	12
1	1	2	3	4	5	6	7	8	9	10	11	12
2	2	4	6	8	10	12	14	16	18	20	22	24
3	3	6	9	12	15	18	21	24	27	30	33	36
4	4	8	12	16	20	24	28	32	36	40	44	48
5	5	10	15	20	25	30	35	40	45	50	55	60
6	6	12	18	24	30	36	42	48	54	60	66	72
7	7	14	21	28	35	42	49	56	63	70	77	84
8	8	16	24	32	40	48	56	64	72	80	88	96
9	9	18	27	36	45	54	63	72	81	90	99	108
10	10	20	30	40	50	60	70	80	90	100	110	120
11	11	22	33	44	55	66	77	88	99	110	121	132
12	12	24	36	48	60	72	84	96	108	120	132	144

Figure 3. Output of figure 2.

```

\def\declarecount {\allocate0\countdef}%
\def\declaredimen {\allocate1\dimendef}%
\def\declareskip {\allocate2\skipdef}%
\def\declaremuskip{\allocate3\muskipdef}%
\def\declarebox {\allocate4\chardef}%
\def\declaretoks {\allocate5\toksdef}%
%
\def\allocate#1#2#3{\relax
  \advance\count1#1 by 1
  \ifnum\count1#1<\count19
  \else
    \errmessage{No room for \string#3!}%
  \fi
  #2#3=\count1#1
}

```

Figure 4. Listing of the declare macros.