The luatexbase-regs package

Manuel Pégourié-Gonnard & Élie Roux
Support: lualatex-dev@tug.org

2010/10/10 v0.3

Abstract

This package extends the register allocation scheme of Plain TeX and LaTeX to take advantage of the increased number of registers available in LuaTeX.

Contents

1 Documentation

Since the Plain TeX and LaTeX formats are both frozen, they fail to take into account the extended resources provided by newer TeX-like engines. This package focuses on the allocation scheme for registers. TeX 82 provides 6 kinds or registers: count, dimen, skip, muskip, box and toks and has 256 registers of each kind. ε-TeX and most of its descendants add one kind of register (marks) and offers $2^{15} = 32768$ of each kind. LuaTeX provides $2^{16} = 65536$ registers of each kind. (It also provides new register-like resources, but this package addresses only the resources inherited from ε-TeX.)

More precisely, luatexbase-regs loads the etex package (or makes sure it is preloaded in the format) and then adapts it to the new limits of LuaTeX. Thus, all macros defined by the etex package are made available (most notably, `\loccount`, `\globcountblk`, `\loccountblk` and alike). However, if a register of some kind has been locally allocated before this package is loaded, then the number of allocatable registers of this kind will not be extended to 65536. To avoid this, load luatexbase-regs earlier.

The Plain TeX and LaTeX formats define a new kind of resource: *inserts* which are merely a family (count, dimen, skip, box) of registers with the same number. Inserts allocation begins at 255 and goes toward 0. Thus we can make room for more inserts by making allocation of count-, dimen-, skip- and box-registers start from 256. With real ε-TeX, it may be a bad idea since registers with index greater than 256 have degraded performance due to implementation details, but with LuaTEX the performance is uniform, so we just do it.

2 Implementation

⟨*texpackage⟩

2.1 Preliminaries

Reload protection, especially for Plain TEX.

```
                \csname lltxb@regs@loaded\endcsname
\expandafter\let\csname lltxb@regs@loaded\endcsname\endinput
```

Catcode defenses.

```
\begingroup
  \catcode123 1 % {
  \catcode125 2 % }
  \catcode 35 6 % #
  \toks0{}%
  \def\x{}%
  \def\y#1 #2 {%
    \toks0\expandafter{\the\toks0 \catcode#1 \the\catcode#1}%
    \edef\x{\x \catcode#1 #2}}%
  \y 123 1  % {
  \y 125 2  % }
  \y  35 6  % #
  \y  10 12 % ^^J
  \y  34 12 % "
  \y  36 3  % $ $
  \y  39 12 % '
  \y  40 12 % (
  \y  41 12 % )
  \y  42 12 % *
  \y  43 12 % +
  \y  44 12 % ,
  \y  45 12 % -
  \y  46 12 % .
  \y  47 12 % /
  \y  60 12 % <
  \y  61 12 % =
  \y  64 11 % @ (letter)
  \y  62 12 % >
  \y  95 12 % _ (other)
  \y  96 12 % `
  \edef\y#1{\endgroup\edef#1{\the\toks0\relax}\x}%
\expandafter\y\csname lltxb@regs@AtEnd\endcsname
```

Package declaration.

```
\begingroup
  \expandafter\ifx\csname ProvidesPackage\endcsname\relax
    \def\x#1[#2]{\immediate\write16{Package: #1 #2}}
  \else
    \let\x\ProvidesPackage
  \fi
\expandafter\endgroup
\x{luatexbase-regs}[2010/10/10 v0.3 Registers allocation for LuaTeX]
```

Make sure LuaTeX is used.

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname RequirePackage\endcsname\relax
  \input ifluatex.sty
\else
  \RequirePackage{ifluatex}
\fi
\ifluatex\else
  \begingroup
    \expandafter\ifx\csname PackageError\endcsname\relax
      \def\x#1#2#3{\begingroup \newlinechar10
        \errhelp{#3}\errmessage{Package #1 error: #2}\endgroup}
    \else
      \let\x\PackageError
    \fi
  \expandafter\endgroup
  \x{luatexbase-attr}{LuaTeX is required for this package. Aborting.}{%
    This package can only be used with the LuaTeX engine^^J%
    (command ‘lualatex’ or ‘luatex’).^^J%
    Package loading has been stopped to prevent additional errors.}
  \lltxb@regs@AtEnd
  \expandafter\endinput
\fi
```

2.2 Ensure etex.sty is loaded

If running LaTeX, load etex.sty. If not, either etex.src was loaded at format generation time, or we cannot do anything.

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname RequirePackage\endcsname\relax \else
  \RequirePackage{etex}[1998/03/26]
\fi
```

To the best of my (mpg) knowledge, all Plain-based formats built with ε-TeX-enabled engines in TeX Live load etex.src. However, let’s be careful and check that etex.sty or etex.src is loaded.

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname et@xins\endcsname\relax
  \begingroup
    \expandafter\ifx\csname PackageWarningNoLine\endcsname\relax
      \def\x#1#2{\begingroup\newlinechar10
        \immediate\write16{Package #1 warning: #2}\endgroup}
    \else
      \let\x\PackageWarningNoLine
    \fi
  \expandafter\endgroup
  \x{luatexbase-regs}{etex macros not loaded!^^J%
    Registers allocation scheme will not be extended.}
\else
```

2.3 Adapt range

First, increase the upper bound for all kinds of registers. Copy code to avoid defining a macro.

```
\ifnum\count270=32768 \count270=65536 \fi
\ifnum\count271=32768 \count271=65536 \fi
\ifnum\count272=32768 \count272=65536 \fi
\ifnum\count273=32768 \count272=65536 \fi
\ifnum\count273=32768 \count273=65536 \fi
\ifnum\count274=32768 \count274=65536 \fi
\ifnum\count275=32768 \count275=65536 \fi
\ifnum\count276=32768 \count276=65536 \fi
```

2.4 Patch macros that used `\mathchardef`

`\box` registers and `\marks` were previously defined using `\mathchardef` since it had the biggest range under ε-TEX (15-bit number). However, this is not enough for LuaTEX's extended registers. Fortunately, `\chardef`'s range is extended, and now large enough, so use it everywhere instead of `\mathchardef`. Do this inside a group and use `\toks0` to store the list of actions.

```
\begingroup \toks0{}
  \def\@namedef #1{\expandafter                  \def\csname#1\endcsname}
  \def\@outerdef#1{\expandafter\outer\expandafter\def\csname#1\endcsname}
```

Notice that the auxiliary macros will automatically expand to the desired level when necessary, see below.

First, here are the definitions from `etex.src`, in a form adapted to our needs.

```
  \def\def@globbox   #1#2{\@outerdef{#1}{\et@xglob 4 \box    #2}}
  \def\def@locbox    #1#2{\@namedef {#1}{\et@xloc  4 \box    #2}}
  \def\def@globmarks #1#2{\@outerdef{#1}{\et@xglob 6 \marks  #2}}
  \def\def@locmarks  #1#2{\@namedef {#1}{\et@xloc  6 \marks  #2}}
  \def\def@et@xgblk#1#2{\@namedef{#1}##1##2##3##4%
    {\et@xchkblk ##1##2{##4}%
      {\allocationnumber=\count 26##1
        \global \advance \count 26##1 by ##4%
        \global #2##3=\allocationnumber
        \wlog {\string ##3=\string ##2blk{\number ##4}
          at \the \allocationnumber}%
        }%
      }}
  \def\def@et@xlblk#1#2{\@namedef{#1}##1##2##3##4%
    {\et@xchkblk ##1##2{##4}%
      {\advance \count 27##1 by -##4%
        \allocationnumber=\count 27##1
        #2##3=\allocationnumber
        \wlog {\string ##3=\string ##2blk{\number ##4}
          at \the \allocationnumber \space (local)%
          }%
        }%
      }}
```

Then, the definitions from `etex.sty` since they are subtly different (`\outer` status, but also optional spaces or = signs).

```
  \def\alt@globbox   #1#2{\@namedef{#1}{\et@xglob 4\box    #2}}
  \def\alt@locbox    #1#2{\@namedef{#1}{\et@xloc  4\box    #2}}
  \def\alt@globmarks #1#2{\@namedef{#1}{\et@xglob 6\marks  #2}}
  \def\alt@locmarks  #1#2{\@namedef{#1}{\et@xloc  6\marks  #2}}
  \def\alt@et@xgblk#1#2{\@namedef{#1}##1##2##3##4%
    {\et@xchkblk##1##2{##4}%
      {\allocationnumber\count26##1%
        \global\advance\count26##1by##4%
        \global#2##3\allocationnumber
        \wlog{\string##3=\string##2blk{\number##4} at
          \the\allocationnumber}%
        }%
      }}
  \def\alt@et@xlblk#1#2{\@namedef{#1}##1##2##3##4%
    {\et@xchkblk##1##2{##4}%
      {\advance\count27##1-##4%
        \allocationnumber\count27##1%
        #2##3\allocationnumber
        \et@xwlog{\string##3=\string##2blk{\number##4} at
          \the\allocationnumber\space(local)}%
        }%
      }}
```

Now, a macro checking the definitions, and making the appropriate re-definition.

```
  \def\check@def#1{%
    \csname def@#1\endcsname{test@#1}\mathchardef
    \expandafter\ifx\csname test@#1\expandafter\endcsname
                    \csname #1\endcsname
      \expandafter\let\csname #1\endcsname\relax
      \toks0\expandafter{\the\toks0\csname def@#1\endcsname{#1}\chardef}
    \else
      \csname alt@#1\endcsname{test@#1}\mathchardef
      \expandafter\ifx\csname test@#1\expandafter\endcsname
                      \csname #1\endcsname
        \toks0\expandafter{\the\toks0\csname alt@#1\endcsname{#1}\chardef}
      \else
        \expandafter\show\csname BAD#1\endcsname
      \fi
    \fi}
```

Now, actually do it.

```
  \check@def{globbox}
  \check@def{locbox}
  \check@def{globmarks}
  \check@def{locmarks}
  \check@def{et@xgblk}
  \check@def{et@xlblk}
\expandafter \endgroup
\the\toks0
```

2.5 Make room for inserts

Finally, make allocation of `\count`, `\dimen`, `skip` and `\box` start with numbers > 255, in order to free the lower numbers for insertions. Be careful with `\new...` macros which are `\outer` in

Plain, since we're in the middle of an `\if` test.

```
  \expandafter\let\csname newcount\endcsname\globcount
  \expandafter\let\csname newdimen\endcsname\globdimen
  \expandafter\let\csname newskip\endcsname\globskip
  \expandafter\let\csname newbox\endcsname\globbox
\fi
```

That's all folks!

```
\lltxb@regs@AtEnd
⟨/texpackage⟩
```

3 Test files

Here we test only the two main formats: Plain TEX (with `etex.src` loaded) and LATEX, both with the LuaTEX engine. Those correspond to the `luatex` and `lualatex` commands in TEX Live.

We want to make sure we can globally and locally allocate 30000 registers of each kind, and still globally allocate 100 `\inserts`. Next we globally allocate a bloc of 3000 registers of each kind, and locally a block of 1000. (Those numbers are not optimal, but they should be enough for testing purposes.)

```
⟨testplain⟩\input luatexbase-regs.sty
⟨testlatex⟩\RequirePackage{luatexbase-regs}
⟨*testplain, testlatex⟩
\def\checkregister#1{%
  \edef\newregister{\expandafter\noexpand\csname new#1\endcsname}%
  \edef\locregister{\expandafter\noexpand\csname loc#1\endcsname}%
  \count0 1
  \loop
    \newregister\dummy
    \locregister\dummy
  \ifnum\count0<30000
    \advance\count0 1
  \repeat}
\checkregister{count}
\checkregister{dimen}
\checkregister{skip}
\checkregister{muskip}
\checkregister{box}
\checkregister{toks}
\checkregister{marks}

\count0 1
\loop \ifnum\count0<100
  \csname newinsert\endcsname\dummy
  \advance\count0 1
\repeat

\globcountblk \dummy{3000}
\globdimenblk \dummy{3000}
\globskipblk  \dummy{3000}
\globmuskipblk\dummy{3000}
\globboxblk   \dummy{3000}
```

```
\globtoksblk  \dummy{3000}
\globmarksblk \dummy{3000}

\loccountblk  \dummy{1000}
\locdimenblk  \dummy{1000}
\locskipblk   \dummy{1000}
\locmuskipblk \dummy{1000}
\locboxblk    \dummy{1000}
\loctoksblk   \dummy{1000}
\locmarksblk  \dummy{1000}
⟨/testplain, testlatex⟩
⟨testplain⟩\bye
⟨testlatex⟩\stop
```