

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 Lua \TeX .

Contents

| | | |
|----------|--|----------|
| 1 | Documentation | 1 |
| 2 | Implementation | 2 |
| 2.1 | Preliminaries | 2 |
| 2.2 | Ensure <code>etex.sty</code> is loaded | 3 |
| 2.3 | Adapt range | 4 |
| 2.4 | Patch macros that used <code>\mathchardef</code> | 4 |
| 2.5 | Make room for inserts | 5 |
| 3 | Test files | 6 |

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 of 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. Lua \TeX 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 Lua \TeX . 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

```
1 (*texpackage)
```

2.1 Preliminaries

Reload protection, especially for Plain TeX.

```
2             \csname lltxb@regs@loaded\endcsname
3 \expandafter\let\csname lltxb@regs@loaded\endcsname\endinput
   Catcode defenses.
4 \begingroup
5   \catcode123 1 % {
6   \catcode125 2 % }
7   \catcode 35 6 % #
8   \toks0{}%
9   \def\x{}%
10  \def\y#1 #2 {%
11    \toks0\expandafter{\the\toks0 \catcode#1 \the\catcode#1}%
12    \edef\x{\x \catcode#1 #2}}%
13  \y 123 1 % {
14  \y 125 2 % }
15  \y 35 6 % #
16  \y 10 12 % ^^J
17  \y 34 12 % "
18  \y 36 3 % $ $
19  \y 39 12 % '
20  \y 40 12 % (
21  \y 41 12 % )
22  \y 42 12 % *
23  \y 43 12 % +
24  \y 44 12 % ,
25  \y 45 12 % -
26  \y 46 12 % .
27  \y 47 12 % /
28  \y 60 12 % <
29  \y 61 12 % =
30  \y 64 11 % @ (letter)
31  \y 62 12 % >
32  \y 95 12 % _ (other)
33  \y 96 12 % `
34  \edef\y#1{\endgroup\edef#1{\the\toks0\relax}\x}%
35 \expandafter\y\csname lltxb@regs@AtEnd\endcsname
   Package declaration.
36 \begingroup
37 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
38   \def\x#1[#2]{\immediate\write16{Package: #1 #2}}
39 \else
40   \let\x\ProvidesPackage
```

```

41 \fi
42 \expandafter\endgroup
43 \x{luatexbase-regs}[2010/10/10 v0.3 Registers allocation for LuaTeX]
    Make sure LuaTeX is used.
44 \begingroup\expandafter\expandafter\expandafter\endgroup
45 \expandafter\ifx\csname RequirePackage\endcsname\relax
46 \input ifluatex.sty
47 \else
48 \RequirePackage{ifluatex}
49 \fi
50 \ifluatex\else
51 \begingroup
52 \expandafter\ifx\csname PackageError\endcsname\relax
53 \def\x#1#2#3{\begingroup \newlinechar10
54 \errhelp{#3}\errmessage{Package #1 error: #2}\endgroup}
55 \else
56 \let\x\PackageError
57 \fi
58 \expandafter\endgroup
59 \x{luatexbase-attr}{LuaTeX is required for this package. Aborting.}{%
60 This package can only be used with the LuaTeX engine^^J%
61 (command 'lualatex' or 'luatex').^^J%
62 Package loading has been stopped to prevent additional errors.}
63 \lltxb@regs@AtEnd
64 \expandafter\endinput
65 \fi

```

2.2 Ensure etex.sty is loaded

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

```

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

```

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

```

70 \begingroup\expandafter\expandafter\expandafter\endgroup
71 \expandafter\ifx\csname et@xins\endcsname\relax
72 \begingroup
73 \expandafter\ifx\csname PackageWarningNoLine\endcsname\relax
74 \def\x#1#2{\begingroup\newlinechar10
75 \immediate\write16{Package #1 warning: #2}\endgroup}
76 \else
77 \let\x\PackageWarningNoLine
78 \fi
79 \expandafter\endgroup
80 \x{luatexbase-regs}{etex macros not loaded!^^J%
81 Registers allocation scheme will not be extended.}
82 \else

```

2.3 Adapt range

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

```
83 \ifnum\count270=32768 \count270=65536 \fi
84 \ifnum\count271=32768 \count271=65536 \fi
85 \ifnum\count272=32768 \count272=65536 \fi
86 \ifnum\count273=32768 \count273=65536 \fi
87 \ifnum\count274=32768 \count274=65536 \fi
88 \ifnum\count275=32768 \count275=65536 \fi
89 \ifnum\count276=32768 \count276=65536 \fi
90 \ifnum\count277=32768 \count277=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.

```
91 \begingroup \toks0{
92   \def\@namedef #1{\expandafter           \def\csname#1\endcsname}
93   \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.

```
94 \def\def@globbox #1#2{\@outerdef{#1}{\et@xglob 4 \box #2}}
95 \def\def@locbox #1#2{\@namedef {#1}{\et@xloc 4 \box #2}}
96 \def\def@globmarks #1#2{\@outerdef{#1}{\et@xglob 6 \marks #2}}
97 \def\def@locmarks #1#2{\@namedef {#1}{\et@xloc 6 \marks #2}}
98 \def\def@et@xgblk#1#2{\@namedef{#1}##1##2##3##4%
99   {\et@xchkblk ##1##2{##4}%
100   {\allocationnumber=\count 26##1
101     \global \advance \count 26##1 by ##4%
102     \global #2##3=\allocationnumber
103     \wlog {\string ##3=\string ##2blk{\number ##4}
104       at \the \allocationnumber}%
105   }%
106   }}
107 \def\def@et@xlblk#1#2{\@namedef{#1}##1##2##3##4%
108   {\et@xchkblk ##1##2{##4}%
109   {\advance \count 27##1 by -##4%
110     \allocationnumber=\count 27##1
111     #2##3=\allocationnumber
112     \wlog {\string ##3=\string ##2blk{\number ##4}
113       at \the \allocationnumber \space (local)%
114     }%
115   }%
116   }}
```

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

```
117 \def\alt@globbox #1#2{\@namedef{#1}{\et@xglob 4\box #2}}
```

```

118 \def\alt@locbox #1#2{\@namedef{#1}{\et@xloc 4\box #2}}
119 \def\alt@globmarks #1#2{\@namedef{#1}{\et@xglob 6\marks #2}}
120 \def\alt@locmarks #1#2{\@namedef{#1}{\et@xloc 6\marks #2}}
121 \def\alt@et@xgblk#1#2{\@namedef{#1}##1##2##3##4%
122   {\et@xchkblk##1##2{##4}%
123   {\allocationnumber\count26##1%
124     \global\advance\count26##1by##4%
125     \global#2##3\allocationnumber
126     \wlog{\string##3=\string##2blk{\number##4} at
127       \the\allocationnumber}%
128   }%
129   }}
130 \def\alt@et@xlblk#1#2{\@namedef{#1}##1##2##3##4%
131   {\et@xchkblk##1##2{##4}%
132   {\advance\count27##1-##4%
133     \allocationnumber\count27##1%
134     #2##3\allocationnumber
135     \et@xwlog{\string##3=\string##2blk{\number##4} at
136       \the\allocationnumber\space(local)}}%
137   }%
138   }}

```

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

```

139 \def\check@def#1{%
140   \csname def@#1\endcsname{test@#1}\mathchardef
141   \expandafter\ifx\csname test@#1\expandafter\endcsname
142     \csname #1\endcsname
143     \expandafter\let\csname #1\endcsname\relax
144     \toks0\expandafter{\the\toks0\csname def@#1\endcsname{#1}\chardef}
145   \else
146     \csname alt@#1\endcsname{test@#1}\mathchardef
147     \expandafter\ifx\csname test@#1\expandafter\endcsname
148       \csname #1\endcsname
149       \toks0\expandafter{\the\toks0\csname alt@#1\endcsname{#1}\chardef}
150     \else
151       \expandafter\show\csname BAD#1\endcsname
152     \fi
153   \fi}

```

Now, actually do it.

```

154 \check@def{globbox}
155 \check@def{locbox}
156 \check@def{globmarks}
157 \check@def{locmarks}
158 \check@def{et@xgblk}
159 \check@def{et@xlblk}
160 \expandafter \endgroup
161 \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.

```
162 \expandafter\let\csname newcount\endcsname\globcount
163 \expandafter\let\csname newdimen\endcsname\globdimen
164 \expandafter\let\csname newskip\endcsname\globskip
165 \expandafter\let\csname newbox\endcsname\globbox
166 \fi
```

That's all folks!

```
167 \lltxb@regs@AtEnd
168 \texpackage)
```

3 Test files

Here we test only the two main formats: Plain \TeX (with `etex.src` loaded) and \LaTeX , both with the Lua \TeX 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.)

```
169 (testplain)\input luatexbase-regs.sty
170 (testlatex)\RequirePackage{luatexbase-regs}
171 (*testplain,testlatex)
172 \def\checkregister#1{%
173   \edef\newregister{\expandafter\noexpand\csname new#1\endcsname}%
174   \edef\locregister{\expandafter\noexpand\csname loc#1\endcsname}%
175   \count0 1
176   \loop
177     \newregister\dummy
178     \locregister\dummy
179     \ifnum\count0<30000
180       \advance\count0 1
181     \repeat}
182 \checkregister{count}
183 \checkregister{dimen}
184 \checkregister{skip}
185 \checkregister{muskip}
186 \checkregister{box}
187 \checkregister{toks}
188 \checkregister{marks}
189
190 \count0 1
191 \loop \ifnum\count0<100
192   \csname newinsert\endcsname\dummy
193   \advance\count0 1
194 \repeat
195
196 \globcountblk \dummy{3000}
197 \globdimenblk \dummy{3000}
198 \globskipblk \dummy{3000}
199 \globmuskipblk \dummy{3000}
200 \globboxblk \dummy{3000}
```

```
201 \globtoksblk \dummy{3000}  
202 \globmarksblk \dummy{3000}  
203  
204 \loccountblk \dummy{1000}  
205 \locdimenblk \dummy{1000}  
206 \locskipblk \dummy{1000}  
207 \locmuskipblk \dummy{1000}  
208 \locboxblk \dummy{1000}  
209 \loctoksblk \dummy{1000}  
210 \locmarksblk \dummy{1000}  
211 </testplain, testlatex>  
212 <testplain>\bye  
213 <testlatex>\stop
```