low level TEX

localboxes

Contents

1 Introduction

The LuaTEX engine inherited a few features from other engines and adding local boxes to paragraphs is one of them. This concept comes from Omega but over time it has been made a bit more robust, also by using native par nodes instead of whatsit nodes that are used to support TEX's extensions. In another low level manual we discuss paragraph properties and these local boxes are also part of that so you might want to catch up on that. Local boxes are stored in an initial par node with an adequate subtype but users wont' notice this (unless they mess around in Lua). The inline par nodes have a different subtype and are injected with the `\localinterlinepenalty`, `\localbrokenpenalty`, `\localleftbox`, `\localrightbox` and LuaMetaTEX specific `\localmiddlebox` primitives. WHen these primitives are used in vertical mode they just set registers.

The original (Omega) idea was that local boxes are used for repetitive punctuation (like quotes) at the left and/or right end of the lines that make up a paragraph. That means that when these primitives inject nodes they actually introduce states so that a stretch of text can be marked.

When this mechanism was cleaned up in LuaMetaTEX I decided to investigate if other usage made sense. After all, it is a feature that introduces some extra code and it then pays of to use it when possible. Among the extensions are a callback that is triggered when the left and right boxes get added and experiments with that showed some potential but in order to retain performance as well as limit extensive node memory usage (par nodes are large) a system of indices was added. All this will be illustrated below. Warning: the mechanism in LuaMetaTEX is not compatible with LuaTEX.

This is a preliminary, uncorrected manual.

2 The basics

This mechanism uses a mix of setting (pseudo horizontal) box registers that get associated with (positions in a) paragraph. When the lines resulting from breaking the list gets packaged into an horizontal (line) box, the local left and right boxes get prepended

and appended to the textual part (inside the left, right and parfills kips and left or right hanging margins). When assigning the current local boxes to the paragraph node(s) references to the pseudo registers are used and the packaging actually copies them. This mix of referencing and copying is somewhat tricky but the engine does it best to hide this for the user.

This mechanism is rather useless when not wrapped into some high level mechanism because by default setting these boxes wipes the existing value. In LuaMetaTEX you can actually access the boxes so prepending and appending is possible but experiments showed that this could come with a huge performance hit when the lists are not cleaned up during a run. This is why we have introduced indices: when you assign local boxes using the index option that specific index will be replaced and therefore we have a more sparse solution. So, contrary to LuaTEX, in LuaMetaTEX the local box registers have a linked lists of local boxes tagged by index. Unless you manipulate in Lua, this is hidden from the user. One can access the boxes from the TEX the but there can be no confusion with LuaTEX here because there we don't have access. This is why usage as in LuaTEX will also work in LuaMetaTEX.

This mechanism obeys grouping as is demonstrated in the next three examples. The first example is:

```
\start
    \dorecurse{10}{test #1.1 }
    \localleftbox{\blackrule[width=2em,color=darkred] }
    \dorecurse{20}{test #1.2 }
    \removeunwantedspaces
    \localrightbox{ \blackrule[width=3em,color=darkblue]}
    \dorecurse{20}{test #1.3 }
\stop
    \dorecurse{20}{test #1.4 }
    % par ends here
```

The next example differs in a subtle way: watch the keep keyword, it makes the setting retain after the group ends.

```
\start
    \start
        \dorecurse{10}{test #1.1 }
        \localleftbox keep {\blackrule[width=2em,color=darkred] }
        \dorecurse{20}{test #1.2 }
        \removeunwantedspaces
        \localrightbox { \blackrule[width=3em,color=darkblue]}
```

```
        \dorecurse{20}{test #1.3 }
    \stop
        \dorecurse{20}{test #1.4 }
\stop
% par ends here
```

The third example has two times `keep`. This option is LuaMetaTEX specific.

```
\start
    \start
        \dorecurse{10}{test #1.1 }
        \localleftbox keep {\blackrule[width=2em,color=darkred] }
        \dorecurse{20}{test #1.2 }
        \removeunwantedspaces
        \localrightbox keep { \blackrule[width=3em,color=darkblue]}
        \dorecurse{20}{test #1.3 }
    \stop
        \dorecurse{20}{test #1.4 }
\stop
% par ends here
```

One (nasty) side effect is that when you set these boxes ungrouped they are applied to whatever follows, which is why resetting them is built in the relevant parts of ConTEXt. The next examples are typeset grouped an demonstrate the use of indices:

```
\dorecurse{20}{before #1 }
\localleftbox{\bf \darkred L 1 }%
\localleftbox{\bf \darkred L 2 }%
\dorecurse{20}{after #1 }
```

before 1 before 2 before 3 before 4 before 5 before 6 before 7 before 8 before 9 before 10 before 11 before 12 before 13 before 14 before 15 before 16 before 17 before 18 before 19 before 20 after 1 after 2 after 3 after 4 after 5 after 6 after 7 after 8 after 9 **L 2** after 10 after 11 after 12 after 13 after 14 after 15 after 16 after 17 after 18 after **L 2** 19 after 20

Indices can be set for both sides:

```
\dorecurse{5}{\localrightbox index #1{ \bf \darkgreen R #1}}%
\dorecurse{20}{before #1 }
\dorecurse{5}{\localleftbox index #1{\bf \darkred L #1 }}%
\dorecurse{20}{after #1 }
```

test 1.1 test 2.1 test 3.1 test 4.1 test 5.1 test 6.1 test 7.1 test 8.1 test 9.1 test 10.1 test 1.2 test 2.2 test 3.2 test 4.2 test 5.2 test 6.2 test 7.2 test 8.2 test 9.2 test 10.2 test 11.2 test 12.2 test 13.2 test 14.2 test 15.2 test 16.2 test 17.2 test 18.2 test 19.2 test 20.2 test 1.3 test 2.3 test 3.3 test 4.3 test 5.3 test 6.3 test 7.3 test 8.3 test 9.3 test 10.3 test 11.3 test 12.3 test 13.3 test 14.3 test 15.3 test 16.3 test 17.3 test 18.3 test 19.3 test 20.3 test 1.4 test 2.4 test 3.4 test 4.4 test 5.4 test 6.4 test 7.4 test 8.4 test 9.4 test 10.4 test 11.4 test 12.4 test 13.4 test 14.4 test 15.4 test 16.4 test 17.4 test 18.4 test 19.4 test 20.4

Example 1

test 1.1 test 2.1 test 3.1 test 4.1 test 5.1 test 6.1 test 7.1 test 8.1 test 9.1 test 10.1 test 1.2 test 2.2 test 3.2 test 4.2 test 5.2 test 6.2 test 7.2 test 8.2 test 9.2 test 10.2 test 11.2 test 12.2 test 13.2 test 14.2 test 15.2 test 16.2 test 17.2 test 18.2 test 19.2 test 20.2 test 1.3 test 2.3 test 3.3 test 4.3 test 5.3 test 6.3 test 7.3 test 8.3 test 9.3 test 10.3 test 11.3 test 12.3 test 13.3 test 14.3 test 15.3 test 16.3 test 17.3 test 18.3 test 19.3 test 20.3 test 1.4 test 2.4 test 3.4 test 4.4 test 5.4 test 6.4 test 7.4 test 8.4 test 9.4 test 10.4 test 11.4 test 12.4 test 13.4 test 14.4 test 15.4 test 16.4 test 17.4 test 18.4 test 19.4 test 20.4

Example 2

test 1.1 test 2.1 test 3.1 test 4.1 test 5.1 test 6.1 test 7.1 test 8.1 test 9.1 test 10.1 test 1.2 test 2.2 test 3.2 test 4.2 test 5.2 test 6.2 test 7.2 test 8.2 test 9.2 test 10.2 test 11.2 test 12.2 test 13.2 test 14.2 test 15.2 test 16.2 test 17.2 test 18.2 test 19.2 test 20.2 test 1.3 test 2.3 test 3.3 test 4.3 test 5.3 test 6.3 test 7.3 test 8.3 test 9.3 test 10.3 test 11.3 test 12.3 test 13.3 test 14.3 test 15.3 test 16.3 test 17.3 test 18.3 test 19.3 test 20.3 test 1.4 test 2.4 test 3.4 test 4.4 test 5.4 test 6.4 test 7.4 test 8.4 test 9.4 test 10.4 test 11.4 test 12.4 test 13.4 test 14.4 test 15.4 test 16.4 test 17.4 test 18.4 test 19.4 test 20.4

Example 3

Figure 1

before 1 before 2 before 3 before 4 before 5 before 6 before 7 **R 1 R 2 R 3 R 4 R 5** before 8 before 9 before 10 before 11 before 12 before 13 before **R 1 R 2 R 3 R 4 R 5** 14 before 15 before 16 before 17 before 18 before 19 before 20 **R 1 R 2 R 3 R 4 R 5** after 1 after 2 after 3 after 4 after 5 after 6 after 7 after 8 after **R 1 R 2 R 3 R 4 R 5** **L 1 L 2 L 3 L 4 L 5** 9 after 10 after 11 after 12 after 13 after 14 **R 1 R 2 R 3 R 4 R 5** **L 1 L 2 L 3 L 4 L 5** after 15 after 16 after 17 after 18 after 19 **R 1 R 2 R 3 R 4 R 5** **L 1 L 2 L 3 L 4 L 5** after 20 **R 1 R 2 R 3 R 4 R 5**

We can instruct this mechanism to hook the local box into the main par node by using the `par` keyword. Keep in mind that these local boxes only come into play when the

lines are broken, so till then changing them is possible.

```
\dorecurse{3}{\localrightbox index #1{ \bf \darkgreen R #1}}%
\dorecurse{20}{before #1 }
\dorecurse{2}{\localleftbox par index #1{\bf \darkred L #1 }}%
\dorecurse{20}{after #1 }
```

L 1 L 2 before 1 before 2 before 3 before 4 before 5 before 6 before 7 before **R 1 R 2 R 3**
L 1 L 2 8 before 9 before 10 before 11 before 12 before 13 before 14 before **R 1 R 2 R 3**
L 1 L 2 15 before 16 before 17 before 18 before 19 before 20 after 1 after 2 **R 1 R 2 R 3**
L 1 L 2 after 3 after 4 after 5 after 6 after 7 after 8 after 9 after 10 after 11 **R 1 R 2 R 3**
L 1 L 2 after 12 after 13 after 14 after 15 after 16 after 17 after 18 after 19 **R 1 R 2 R 3**
L 1 L 2 after 20 **R 1 R 2 R 3**

3 The interface

The interface described here is experimental.

Because it is hard to foresee if this mechanism will be used at all the ConTEXt interface is somewhat low level: one can build functionality on top of it. In the previous section we saw examples of local boxes being part of the text but one reason for extending the interface was to see if we can also use this engine feature for efficiently placing marginal content.

```
\definelocalboxes
  [lefttext]
  [location=lefttext,width=3em,color=darkblue]
\definelocalboxes
  [lefttextx]
  [location=lefttext,width=3em,color=darkblue]

\definelocalboxes
  [righttext]
  [location=righttext,width=3em,color=darkyellow]
\definelocalboxes
  [righttextx]
  [location=righttext,width=3em,color=darkyellow]
```

The order of definition matters! Here the x variants have a larger index number. There can (currently) be at most 256 indices. The defined local boxes are triggered with `\localbox`:

```
\startnarrower
\dorecurse{20}{before #1 }%
\localbox[lefttext]{[L] }%
\localbox[lefttextx]{[LL] }%
\localbox[righttext]{ [RR]}%
\localbox[righttextx]{ [R]}%
\dorecurse{20}{ after #1}%
\stopnarrower
```

Watch how we obey the margins:

> before 1 before 2 before 3 before 4 before 5 before 6 before 7 before 8 before 9 before 10 before 11 before 12 before 13 before 14 before 15 before 16 before 17 before 18 before 19 before 20 after 1 after 2 after 3 after 4 after 5 after [RR] [R] [L] [LL] 6 after 7 after 8 after 9 after 10 after 11 after 12 after 13 after [RR] [R] [L] [LL] 14 after 15 after 16 after 17 after 18 after 19 after 20 [RR] [R]

Here these local boxes have dimensions. The predefined margin variants are virtual. Here we set up the style and color:

```
\setuplocalboxes
  [leftmargin]
  [style=\bs,
   color=darkgreen]
\setuplocalboxes
  [rightmargin]
  [style=\bs,
   color=darkred]

\dorecurse{2}{
    \dorecurse{10}{some text #1.##1 }%
    KEY#1.1%
    \localmargintext[leftmargin]{L #1.1}%
    \localmargintext[rightmargin]{R #1.1}%
    \dorecurse{10}{some text #1.##1 }%
    KEY#1.2%
    \localmargintext[leftmargin]{L #1.2}%
    \localmargintext[rightmargin]{R #1.2}%
    \dorecurse{10}{some text #1.##1 }%
    \blank
}
```

You can also use `leftedge` and `rightedge` but using them here would put them outside the page.

some text 1.1 some text 1.2 some text 1.3 some text 1.4 some text 1.5 some text 1.6
L 1.1 some text 1.7 some text 1.8 some text 1.9 some text 1.10 KEY1.1some text 1.1 some ***R 1.1***
text 1.2 some text 1.3 some text 1.4 some text 1.5 some text 1.6 some text 1.7 some
L 1.2 text 1.8 some text 1.9 some text 1.10 KEY1.2some text 1.1 some text 1.2 some text 1.3 ***R 1.2***
some text 1.4 some text 1.5 some text 1.6 some text 1.7 some text 1.8 some text 1.9
some text 1.10

some text 2.1 some text 2.2 some text 2.3 some text 2.4 some text 2.5 some text 2.6
L 2.1 some text 2.7 some text 2.8 some text 2.9 some text 2.10 KEY2.1some text 2.1 some ***R 2.1***
text 2.2 some text 2.3 some text 2.4 some text 2.5 some text 2.6 some text 2.7 some
L 2.2 text 2.8 some text 2.9 some text 2.10 KEY2.2some text 2.1 some text 2.2 some text 2.3 ***R 2.2***
some text 2.4 some text 2.5 some text 2.6 some text 2.7 some text 2.8 some text 2.9
some text 2.10

In previous examples you can see that setting something at the left will lag behind so deep down we use another trick here: `\localmiddlebox`. When these boxes get placed a callback can be triggered and in ConTEXt we use that to move these middle boxes to the margins.

Next we implement line numbers. Watch out: this will not replace the existing mechanisms, it's just an alternative as we have alternative table mechanisms. We have a repertoire of helpers for constructing the result:

```
\definelocalboxes
  [linenumberleft]
  [command=\LeftNumber,
   location=middle,
   distance=\leftmargindistance,
   width=3em,
   style=\bs,
   color=darkred]

\definelocalboxes
  [linenumberright] % [linenumberleft]
  [command=\RightNumber,
   location=middle,
   distance=\rightmargindistance,
   width=3em,
   style=\bf,
```

```
   color=darkgreen]

\definecounter[MyLineNumberL]
\definecounter[MyLineNumberR]

\setupcounter
  [MyLineNumberL]
  [numberconversion=characters]

\setupcounter
  [MyLineNumberR]
  [numberconversion=romannumerals]

\def\LineNumberL
  {\incrementcounter[MyLineNumberL]%
   \convertedcounter[MyLineNumberL]}

\def\LineNumberR
  {\incrementcounter[MyLineNumberR]%
   \convertedcounter[MyLineNumberR]}

\protected\def\LeftNumber
  {\setbox\localboxcontentbox\hbox
     to \localboxesparameter{width}
     {(\LineNumberL\hss\strut)}%
   \localmarginlefttext\zeropoint}

\protected\def\RightNumber
  {\setbox\localboxcontentbox\hbox
     to \localboxesparameter{width}
     {(\strut\hss\LineNumberR)}%
   \localmarginrighttext\zeropoint}

\localbox[linenumberleft]{}%
\localbox[linenumberright]{}%
\dorecurse{2}{
    \samplefile{tufte}
    \par
}
\resetlocalbox[linenumberleft]%
\resetlocalbox[linenumberright]%
```

We use our tufte example to illustrate the usage:

The interface

(a) We thrive in information–thick worlds because of our marvelous and everyday capacity (i)
(b) to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthe- (ii)
(c) size, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, (iii)
(d) list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeon- (iv)
(e) hole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, aver- (v)
(f) age, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip (vi)
(g) through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, (vii)
(h) winnow the wheat from the chaff and separate the sheep from the goats. (viii)

(i) We thrive in information–thick worlds because of our marvelous and everyday capacity (ix)
(j) to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthe- (x)
(k) size, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, (xi)
(l) list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeon- (xii)
(m) hole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, aver- (xiii)
(n) age, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip (xiv)
(o) through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, (xv)
(p) winnow the wheat from the chaff and separate the sheep from the goats. (xvi)

For convenience we support ranges like this (we've reset the line number counters here):

```
\startlocalboxrange[linenumberleft]%
\startlocalboxrange[linenumberright]%
\dorecurse{2}{
    \samplefile{tufte}
    \par
}
\stoplocalboxrange
\stoplocalboxrange
```

(a) We thrive in information–thick worlds because of our marvelous and everyday capacity (i)
(b) to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthe- (ii)
(c) size, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, (iii)
(d) list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeon- (iv)
(e) hole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, aver- (v)
(f) age, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip (vi)
(g) through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, (vii)
(h) winnow the wheat from the chaff and separate the sheep from the goats. (viii)

(i) We thrive in information–thick worlds because of our marvelous and everyday capacity (ix)
(j) to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthe- (x)
(k) size, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, (xi)

(l) list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeon- (xii)
(m) hole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, aver- (xiii)
(n) age, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip (xiv)
(o) through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, (xv)
(p) winnow the wheat from the chaff and separate the sheep from the goats. (xvi)

4 The helpers

For the moment we have these helpers:

```
\localboxindex             integer
\localboxlinenumber        integer
                           \localboxlinewidthdimension
\localboxlocalwidth        dimension
\localboxprogress          dimension
\localboxleftoffset        dimension
\localboxrightoffset       dimension
                           \localboxleftskip dimension
\localboxrightskip         dimension
\localboxlefthang          dimension
\localboxrighthang         dimension
                           \localboxindent   dimension
\localboxparfillleftskip   dimension
\localboxparfillrightskip  dimension
\localboxovershoot         dimension
```

The progress and offsets are accumulated values of the normalized indent, hangs, skips etc. The line number is the position in the paragraph. In the callback we set the box register `\localboxcontentbox` and use it after the command has been applied. In the line number example you can see how we set its final content, so these boxes are sort of dynamic. Normally in the middle case no content is passed and in the par builder a middle is not taken into account when calculating the line width.

4 Colofon

Author	Hans Hagen
ConTEXt	2023.04.27 17:04
LuaMetaTEX	2.1008
Support	www.pragma-ade.com
	contextgarden.net