

LuaTeX-ja における縦組

北川 弘典

(LuaTeX-ja プロジェクト)

2014 年 11 月 8 日

TeX ユーザの集い 2014

- LuaTeX 上で日本語組版を行うための
マクロパッケージ.
- pTeX から多大な影響を受けているが,
100%互換ではない
- 複数人が参加. 公式ページは
<http://www.sourceforge.jp/projects/luatex-ja/wiki/FrontPage>

■ 組方向への対応

- サポートする組方向
- “primitive” レベル
- L^AT_EX
- fontspec



■ 実装と問題点

- LuaT_EX 本体の「組方向」
- 大まかな実装
- 和文文字の出力

組方向は各ボックスの先頭部で変更可能.

	命令	フォント
横組	<code>\yoko</code>	横組用
縦組	<code>\tate</code>	縦組用
縦数式	(縦中数式)	横組用

pTeX では、和文文字には 16 (漢字)・17 (カナ)・18 (その他記号) の値が `\kcatcode` として **単位で**割り振られており、……

$$\text{Gauss 積分} \int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}$$

ちょっと「チエツク」したら？

組方向は各ボックスの先頭部で変更可能.

	命令	フォント
横組	<code>\yoko</code>	横組用
縦組	<code>\tate</code>	縦組用
縦数式	(縦中数式)	横組用
dtou 方向	<code>\dtou</code>	横組用

dtou 方向は「隠し組方向」だが、それについてはまた別の機会に.

pTeX では、和文字には 16 (漢字)・17 (カナ)・18 (その他記号) の値が `\kcatcode` として **☒** **単位**で割り振られており、……

$$\text{Gauss 積分} \int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}$$

ちょっと「チェック」したら？

組方向は各ボックスの先頭部で変更可能.

	命令	フォント	
横組	<code>\yoko</code>	横組用	
縦組	<code>\tate</code>	縦組用	
縦数式	<code>\utod</code>	横組用	「縦数式ディレクションに移行」する <code>\utod</code> を新設
dtou 方向	<code>\dtou</code>	横組用	dtou 方向の存在とサポートを明文化

pTeX で通る次のコードはどうしようかな.

```
\vbox{\yoko\noindent
  \tate\indent 「あっ」 xyz}
```

組方向判定には専用の if 文を用いる

```
\ifydir  
  (横組時の処理)  
\else % \dtou は今は考えないことにして……  
  \ifmdir  
    (縦数式ディレクションでの処理)  
  \else  
    (縦組時の処理)  
  \fi  
\fi
```

~~組方向判定には専用の if 文を用いる~~

→ direction パラメータを使い, `\ifnum` で判定

```
\ifnum \ltjgetparameter{direction}=4
```

(横組時の処理)

```
\else % \dtou は今は考えないことにして……
```

```
\ifnum \ltjgetparameter{direction}=11
```

(縦数式ディレクションでの処理)

```
\else
```

(縦組時の処理)

```
\fi
```

```
\fi
```

横組	4
縦組	3
縦数式	11
dtou 方向	1

横組

縦組

欧文
和文

yalbaselineshift
yjabaselineshift

talbaselineshift
tjabaselineshift

横組

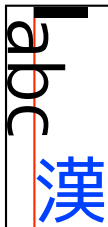
縦組

欧文 yalbaselineshift
和文 yjabaselineshift

talbaselineshift
tjabaselineshift

欧文側の補正は， 予稿時 (9/10) より改善

```
\fboxsep=0pt\fbox{\hbox{\tate  
  \ltjsetParameter{  
    talbaselineshift=5pt,  
    tjabaselineshift=-0.5\zh  
  }  
  \rule{2pt}{10pt}abc 漢}}
```



ベースライン

\wd 他の値は, 現在の組方向に依存する

```
\setbox42=\hbox{\yoko  
  \vrule width 13pt height 8pt depth 2pt}  
\hbox{\yoko\the\wd42}% --> 13pt  
\hbox{\tate\the\wd42}% --> 10pt
```

\wd 他の値は，現在の組方向に依存するしない

```
\setbox42=\hbox{\yoko
  \vrule width 13pt height 8pt depth 2pt}
\hbox{\yoko\the\wd42}% --> 13pt
\hbox{\tate\the\wd42}% --> 10pt13pt
```

組方向に依存したボックスの寸法を使う場合は，

取得 `\ltjgetwd{42}` (内部長さ)

設定 `\ltjsetwd42=17.01pt`

```
\font\HOG E=KozGoPr6N-Regular:%  
-kern;jfm=ujisv
```

```
\HOG E\hbox{\tate\vrule\  
「あっ」と驚く\vrule }
```

縦組用フォントでは明示的な指定なし
でも

vrt2 feature を有効化.
(-vert or -vrt2 で無効化)

「あっ」と驚く

```
\tfont\HOG E=KozGoPr6N-Regular:%  
-kern;jfm=propv;+vpa l  
% プロポーショナル組  
\HOG E\hbox{\tate\vrule\  
「あっ」と驚く\vrule }
```

- x 方向の補正については未考慮.
- (palt + kern), vpa l + vk rn はまだ問題あり (luaotfload 更新待ち)

「あっ」と驚く

	pL ^A T _E X	LuaL ^A T _E X-ja
エンコーディング	JY1, JT1	JY3, JT3
縦組クラス	tarticle.cls tbook.cls treport.cls	ltj tarticle.cls ltj tbook.cls ltj treport.cls
ロード ↓		
縦組拡張マクロ集	plext.sty	lltjext.sty

lltjext.sty での変更は組方向オプション関連が主.

対象 表組, minipage, picture 環境など

- 指定値
- `<y>` (横組), `<t>` (縦組)
 - `<z>` (周囲が縦組のとき,
縦数式ディレクションに移行)

対象 表組, minipage, picture 環境など

- 指定値
- `<y>` (横組), `<t>` (縦組)
 - `<z>` (周囲が縦組のとき,
縦数式ディレクションに移行)
 - `<d>` (dtou 方向), `<u>` (縦数式)

対象 表組, `minipage`, `picture` 環境など

- 指定値
- `<y>` (横組), `<t>` (縦組)
 - `<z>` (周囲が縦組のとき,
縦数式ディレクションに移行)
 - `<d>` (`dtou` 方向), `<u>` (縦数式)

仕様変更 `[t]`, `[b]` の挙動を簡素化

`[t]` → `\vtop`, `[c]` → `\vcenter`, `[b]` → `\vbox`

`lltjext.sty` の有無で表組の垂直位置が変わらないように

```
\jfontspec[
  YokoFeatures={Color=00007F},
  TateFeatures={CJKShape=NLC},
  TateFont=KozGoPr6N-Medium,
]{KozMinPr6N-Regular}
\parbox<y>{3\zw}{葛城市}
\parbox<t>{3\zw}{葛飾区}
```

葛城市

葛
飾
区

```
\jfontspec[
  YokoFeatures={Color=00007F},
  TateFeatures={CJKShape=NLC},
  TateFont=KozGoPr6N-Medium,
]{KozMinPr6N-Regular}
\parbox<y>{3\zw}{葛城市}
\parbox<t>{3\zw}{葛飾区}
```

横組用フォントのみ
有効な feature 達

縦組用フォントのみ
有効な feature 達

縦組用和文フォント
の指定

葛城市	葛 飾 区
-----	-------------

■ 組方向への対応

- サポートする組方向
- “primitive” レベル
- L^AT_EX
- fontspec



■ 実装と問題点

- LuaT_EX 本体の「組方向」
- 大まかな実装
- 和文文字の出力

LuaTeX 本体では Ω スタイルの組方向として次の4つが使用可能：

- **TLT**……通常**の左横書き**
- TRT……右横書き
- LTL……モンゴル文字用
- RTT……「CJK 縦書き用」

The quick brown fox
jumps over the lazy
dog.TLT

LuaTeX 本体では Ω スタイルの組方向として次の4つが使用可能：

- TLT……通常の左横書き
- TRT……右横書き
- LTL……モンゴル文字用
- **RTT……「CJK 縦書き用」**

the lady dog	fox : s re	The dog kick down
TLT		

LuaTeX 本体では Ω スタイルの組方向として次の4つが使用可能：

- TLT……通常の左横書き
- TRT……右横書き
- LTL……モンゴル文字用
- **RTT……「CJK 縦書き用」**

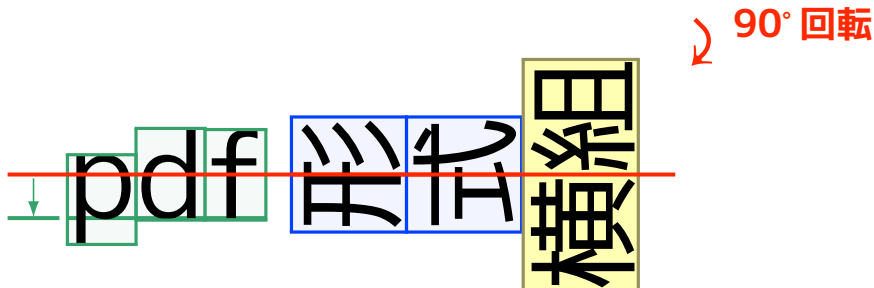
the fox The
land jump quick

だが、RTT は日本語縦組向きではない

(欧文文字の配置, dtou 方向の対応物がない)

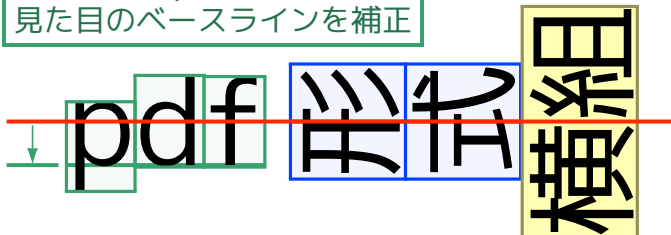
er
wn

全体を TLT で組み，出力時にボックスを回転



全体を TLT で組み，出力時にボックスを回転

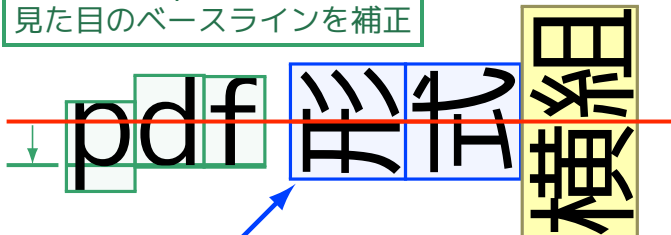
欧文文字は `yoffset` を調整し
見た目のベースラインを補正



90° 回転

全体を TLT で組み，出力時にボックスを回転

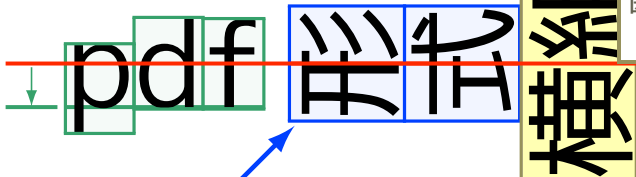
欧文文字は `yoffset` を調整し
見た目のベースラインを補正



和文文字は（非効率だが）
1文字ずつ回転し，
ボックスにカプセル化

全体を TLT で組み，出力時にボックスを回転

欧文文字は `yoffset` を調整し
見た目のベースラインを補正

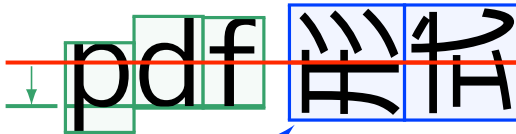


異方向ボックスは
別のボックスで
囲んで整合させる
(`dir_box`)

和文文字は（非効率だが）
1文字ずつ回転し，
ボックスにカプセル化

全体を TLT で組み，出力時にボックスを回転

欧文文字は `yoffset` を調整し
見た目のベースラインを補正



和文文字は（非効率だが）
1文字ずつ回転し，
ボックスにカプセル化

異方向ボックスは
別のボックスで
囲んで整合させる
(`dir_box`)



`dir_box` の処理のため
`\raise` 類，`\vcenter`，
`\lastbox` を再定義

各ボックス・リストの組方向用フィールド (pTeX)

~~各ボックス・リストの組方向用フィールド~~

LuaTeX-ja ではボックス・リストの「先頭」に
組方向格納用 `whatsit` ノードを作成

~~各ボックス・リストの組方向用フィールド~~

LuaTeX-ja ではボックス・リストの「先頭」に
組方向格納用 `whatsit` ノードを作成

- `\everyhbox`, `\everyvbox` を利用
(`everyhook` パッケージが使える時はそっちで)
- `whatsit` に起因する**予想外のバグの可能性**

~~各ボックス・リストの組方向用フィールド~~

LuaTeX-ja ではボックス・リストの「先頭」に
組方向格納用 `whatsit` ノードを作成

- `\everyhbox`, `\everyvbox` を利用
(`everyhook` パッケージが使える時はそっちで)
- `whatsit` に起因する**予想外のバグの可能性**
- 作成処理の実装により, **LuaTeX rev 5021 以前**
(**含 TeX Live 2014**) では**強制終了の可能性**

なぜ縦組で和文文字は「1文字ずつ回転」させる？

∵ LuaTeX で TrueType/OpenType フォントを使うと自動的に **Identity-H エンコーディング** になるから.

- フォント非埋め込みでも同じ状況
- Font Descriptor flag は 4 (symbolic, sans serif) に固定

なぜ縦組で和文文字は「1文字ずつ回転」させる？

∵ LuaTeX で TrueType/OpenType フォントを使うと自動的に **Identity-H エンコーディング** になるから.

- フォント非埋め込みでも同じ状況
- Font Descriptor flag は 4 (symbolic, sans serif) に固定

→ 縦組時には pdf からのテキスト抽出が
使い物にならない

LuaTeX-ja で縦組が可能になった.

- 命令名称の変更を除けば,
pL^AT_EX とあまり変わらない使い勝手
 - `zw` → `\zw` など
 - `\wd` → `\ltjgetwd`, `\ltjsetwd`
 - `fontspec` の縦組サポートも.
- **T_EX Live 2014 では動かない (バイナリの関係)**
W32T_EX の `luatex-dev` なら良いはず
- 縦組中の和文文字の出力などでまだ詰めが必要.