

Experimental p_{La}T_EX 2_ε

Japanese T_EX Development Community

February 9, 2020

注意: これは Experimental な p_{La}T_EX 2_ε、すなわち unstable な実験的 p_{La}T_EX コードを提供するパッケージです。

1 このパッケージの目的

コードの不用意な改変は即エンバグにつながり、利用者の多い p_{La}T_EX や up_{La}T_EX では特に影響が大きいと思われます。その一方で、unstable なものなるべく手軽にテストして頂きたいとも考えます。

このパッケージ `exppl2e.sty` は、カーネル (stable) に将来含めることを想定した unstable な実験的コードを配布することを目的に作成しました。テストをよろしくお願いします。

2 実験的コードの読みこみかた

デフォルトの配布では、実験的なコードは無効化されています。実験的な p_{La}T_EX 2_ε を試したい場合は、以下のいずれかの方法を使います：

2.1 少しだけ試してみたい場合

パッケージ `exppl2e.sty` を読み込みます。ただし、`\usepackage` 命令を使うのではなく、文書クラスより前に読み込んでおくのが無難です。

```
\RequirePackage{exppl2e}
\documentclass{article}
```

2.2 常に実験的コードを使用したい場合

このパッケージと一緒にインストールされる新しい p_{La}T_EX は、お使いの `platex` などのプログラムが見つけることのできる場所（簡単なのはカレントディレクトリ、あるいは `$TEXMFLOCAL/tex` 以下の適切な場所）に `platex.cfg` というファイルがあれば

ば、起動時にそれを読み込みます。この機能を利用すると、以下の内容の `platex.cfg` を用意しておくだけで、自動的に毎回 `exppl2e.sty` が読み込まれます。

```
\RequirePackage{exppl2e}
```

3 このドキュメントについて

コミュニティ版 p \LaTeX が配布するほかの `sty` ファイルとは異なり、実質的には `exppl2e.sty` は `dtx` ファイルと同等です。すなわち、コードと一緒に `dtx` 互換ドキュメントが含まれています。このドキュメントを組版するには

```
# platex exppl2e.sty
```

を実行します。

4 コード

ここから p \LaTeX 2 ϵ の experimental コード本体です。

5 改行

強制改行 `\` と `\par` が連続した場合の挙動については以下のとおり保留中。参考：
GitHub:texjporg/platex#27

`\@gnewline` 日本語 \TeX の行頭禁則処理は、禁則対象文字の直前に、`\prebreakpenalty` で指定されたペナルティの値を挿入することで行なっています。ところが、改行コマンドは負のペナルティの値を挿入することで改行を行ないます。そのために、禁則ペナルティの値が 10000 の文字の直後では、ペナルティの値が相殺され、改行することができません。

```
あいうえお \\  
! かきくけこ
```

したがって、`\newline` マクロに `\mbox{}` を入れることによって、`\newline` マクロのペナルティ -10000 と行頭文字のペナルティ 10000 が加算されないようにします。`\` は `\newline` マクロを呼び出しています。

なお、`\newline` マクロは `ltspace.dtx` で定義されています。

\LaTeX <1996/12/01>で改行マクロが変更され、`\` が `\newline` を呼び出さなくなったため、変更された改行マクロに対応しました。`\null` の挿入位置は同じです。`ltspace.dtx` の定義を上記に合わせて、定義しなおしました。

日本語 $T_{\text{E}}X$ 開発コミュニティによる補足：アスキーによる $\text{p}T_{\text{E}}X$ では、行頭禁則文字の直前で `\\` による強制改行を行えるようにするという目的で `\null` を `\@gnewline` マクロ内に挿入していました。しかし、これでは `\\\par` と書いた場合に Underfull 警告が出なくなっています (`tests/newline_par.tex` を `latex` と `platex` で処理してみてください)。

もし `\null` の代わりに `\hskip\z@` を挿入すれば、 $\text{L}T_{\text{E}}X$ と同様に Underfull 警告を出すことができます。ただし、`\null` を挿入した場合と異なり、強制改行後の行頭に JFM グルーが入らなくなります。これはむしろ、奥村さんの `jsclasses` で行頭を天ツキに直しているのと同じですが、 $\text{p}T_{\text{E}}X$ としては挙動が変化してしまいますので、現時点では `\null` \rightarrow `\hskip\z@` への変更を見送っています。

もし変更するならば、以下のコードを有効にします。

```

1 <platexrelease>\plIncludeInRelease{???/??/?}{\@gnewline}
2 <platexrelease>                                {Restore Underfull warning for |\\\par|}%
3 <*plcore | platexrelease>
4 %\def\@gnewline #1{%
5 %  \ifvmode
6 %    \@nolnerr
7 %  \else
8 %    \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break \hskip \z@
9 %    \ignorespaces
10 %  \fi}
11 </plcore | platexrelease>
12 <platexrelease>\plEndIncludeInRelease
13 <platexrelease>\plIncludeInRelease{0000/00/00}{\@gnewline}
14 <platexrelease>                                {Restore Underfull warning for |\\\par|}%
15 <platexrelease>\def\@gnewline #1{%
16 <platexrelease>  \ifvmode
17 <platexrelease>    \@nolnerr
18 <platexrelease>  \else
19 <platexrelease>    \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break \null
20 <platexrelease>    \ignorespaces
21 <platexrelease>  \fi}
22 <platexrelease>\plEndIncludeInRelease

```

`\linebreak` の挙動の修正は、バグが相次いだため取りやめています。

6 相互参照

$\text{p}T_{\text{E}}X 2_{\epsilon}$ カーネル (2017/10/28) の修正に加え、以下の修正を検討中ですが、現時点では保留しています。参考：GitHub:texjporg/platex#57

`\@setref@` さらに、参照した結果が「空」の場合

```
\documentclass{article}
```

```

\pagenumbering{gobble}
\begin{document}
\pageref{a}\label{a}
\end{document}

```

に相互参照が収束しなくなる¹のを防ぐため、水平モード以外ではやはり `\null` を発行してみます。

```
23 %\def\@setref@{\ifhmode\spacefactor\@m\else\null\fi}
```

このコードは、`emath` の `\marusuuref` のような「`\ref` を使って参照先の番号だけを取得する」というマクロの動作に（垂直モードで使う場合など）影響するため、ひとまずコメントアウトしておきます。

7 空のページ

現時点では保留しています。参考：GitHub:texjporg/platex#78

$\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X} 2_{\epsilon}$ 2017/01/01 以降では、例えば

```

\documentclass{tarticle}
\begin{document}
\begin{figure}
\end{figure}
\clearpage
a
\end{document}

```

のようになると「空のフロート」だけの空白ページを発生させることができます。この場合、縦組クラスを用いているとフッタが持ち上がります。

なお、 $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X} 2_{\epsilon}$ 2015/01/01～2016/03/31 patch level 3 では

```
! Output loop---100 consecutive dead cycles.
```

のエラーが出て、それ以前では空白ページは発生しません。

フロートでない空白ページでページフッタが持ち上がる問題へは、`\@makecol` の `\iftdir\vbox{\hskip\z@\fi}` で対処してありますが、フロートがある空白ページでは `\@makecol` が呼ばれないことから対策が効いていません。ただ、フロートが空である状況は元々不自然なので、現在はサポートしていません。

もし対策することになれば、下記のコードを使用します。

`\@vtryfc` `ltoutput.dtx` で定義されている `\@vtryfc` に `\iftdir\hskip\z@\fi` を追加してあります。

¹奇数回コンパイルすると Output written on test.dvi (1 page, 212 bytes). となり、偶数回コンパイルすると No pages of output. となります。

```

24 \def\@vtryfc #1{%
25   \global\setbox\@outputbox\vbox{\iftdir\hskip\z@\fi}%
26   \let\@elt\@wtryfc
27   \@flsucceed
28   \global\setbox\@outputbox \vbox to\@colht{%
29     \vskip \@fptop
30     \vskip -\@fpsep
31     \unvbox \@outputbox
32     \vskip \@fpbot}%
33   \let\@elt\relax
34   \xdef #1{\@failedlist\@flfail}%
35   \xdef\@freelist{\@freelist\@flsucceed}}

```

8 支柱

`\strut` `\strut` を縦数式ディレクションで使った場合への修正。

```

36 %% \strut の改良版
37 \DeclareRobustCommand\strut{\relax
38   \iftdir
39     \ifmdir
40       %% [縦数式ディレクション]
41       %% テキストモードの場合は \zstrutbox でなければならない気がする。
42       %% (下線 \underline で使う場合に横組の 90 度回転にならない)
43       %% 数式モードの場合は \zstrutbox に変えとかえってバランスが崩れるが,
44       %% これは \copy したボックスにベースライン補正が効かないためであり,
45       %% その分 \lower で下げる。
46       \ifmmode \lower \tbaselineshift \copy \zstrutbox \else \unhcopy \zstrutbox \fi
47     \else
48       %% [縦ディレクション]
49       %% テキストモードの場合は \tstrutbox で良い。
50       %% 数式モードに入ると自動的に [縦数式ディレクション] に入るので分岐不要。
51       \unhcopy \tstrutbox
52     \fi
53   \else
54     %% [横ディレクション]
55     %% テキストモードの場合は \ystrutbox で良い。
56     %% 数式モードではベースライン補正分だけ \lower で下げる。
57     \ifmmode \lower \ybaselineshift \copy \ystrutbox \else \unhcopy \ystrutbox \fi
58   \fi}
59 %% \strutbox は \zstrutbox を使うとまずそうなので 2017-04-08 のままとする。
60 %% 以下のコードは使わない。
61 %\def\strutbox{%
62 %  \iftdir \ifmdir \zstrutbox \else \tstrutbox \fi
63 %  \else \ystrutbox \fi}

```

`\strutbox` の変更は 2017/04/08 に導入済み。

9 \verb 先頭の合字抑制と \xkanjiskip

`\do@noligs` >などの一部の文字について、`\xspacecode{>=3}`としたときに`\texttt{>}`では前後に`\xkanjiskip`由来のアキが入るのに、`\verb+>+`では後ろにしかアキが入らないという現象に対処します。

元の定義は `lrmiscen.dtx` を参照してください。p \LaTeX では、`\kern\z@`を`\vadjust{}`に置き換えることで「合字処理を抑制」かつ「和欧文間スペースの挿入処理は透過」を実現します。(Issue #87)

```
64 \def\do@noligs#1{%
65   \catcode'#1\active
66   \begingroup
67     \lccode'\~'#1\relax
68     \lowercase{\endgroup\def~{\leavevmode\vadjust{}}\char'#1}}
```

10 NFSS の size function の日本語化

`\inlist@` `\do@subst@correction` の日本語化に必要なので、比較する際に引数・リストとも`\detokenize`によって文字列化するようにしました。

```
69 \def\inlist@#1#2{%
70   \edef\reserved@a{%
71     \unexpanded{\def\in@##1<}%
72     \detokenize{#1}%
73     \unexpanded{>##2##3\in@{\ifx\in@##2\in@false\else\in@true\fi}\in@}%
74     \detokenize{#2}%
75     \unexpanded{<}%
76     \detokenize{#1}%
77     \unexpanded{>\in@\in@}}%
78   \reserved@a}
```

`\do@subst@correction` `\font` は欧文フォントを返すため、 \LaTeX の元の `\do@subst@correction` は和文フォントに対して使えませんので、和文に対応させます。

`\pltx@do@subst@correction@al` オリジナルの \LaTeX の定義は、欧文用として使います。

```
\pltx@do@subst@correction@yoko
\pltx@do@subst@correction@tate
79 \def\pltx@do@subst@correction@al{%
80   \xdef\subst@correction{%
81     \font@name
82     \global\expandafter\font
83     \csname \curr@fontshape/\f@size\endcsname
84     \noexpand\fontname\font
85     \relax}%
86   \aftergroup\subst@correction
87 }
```

和文横組用と和文縦組用の定義では、それぞれ `\jfont` と `\tfont` を使います。

```
88 \def\pltx@do@subst@correction@yoko{%
```

```

89      \xdef\subst@correction{%
90      \font@name
91      \global\expandafter\jfont
92      \csname \curr@fontshape/\f@size\endcsname
93      \noexpand\fontname\jfont
94      \relax}%
95      \aftergroup\subst@correction
96 }
97 \def\pltx@do@subst@correction@tate{%
98      \xdef\subst@correction{%
99      \font@name
100     \global\expandafter\tfont
101     \csname \curr@fontshape/\f@size\endcsname
102     \noexpand\fontname\tfont
103     \relax}%
104     \aftergroup\subst@correction
105 }

```

そして、エンコーディングに応じて 3 つの命令を使い分けます。

```

106 \def\do@subst@correction{%
107   \edef\tmp@item{{\f@encoding}}%
108   \expandafter\expandafter\expandafter
109   \inlist@\expandafter\tmp@item\expandafter{\kyenc@list}%
110   \ifin@\pltx@do@subst@correction@yoko
111   \else
112     \expandafter\expandafter\expandafter
113     \inlist@\expandafter\tmp@item\expandafter{\ktenc@list}%
114     \ifin@\pltx@do@subst@correction@tate\else
115     \pltx@do@subst@correction@al
116   \fi
117   \fi
118 }

```

11 脚注の合印直後での改行を許可

2016/09/03 に導入済み。

12 e-pTeX での FAM256 パッチの利用

2016/11/29 に導入済み。

13 脚注とボトムフロートの順序および垂直位置

2017/04/08 に導入済み。

14 `\textunderscore` のベースライン補正

2017/04/08 に導入済み。

15 `verbatim` とハイフネーション

2017/04/08 に導入済み。

16 `\verb` の冒頭のスペース

2017/10/28 に導入済み。

17 `tabbing` 環境の行冒頭の JFM グルー

2017/10/28 に導入済み。

18 `\inhibitglue` の簡略形

2017/10/28 に導入済み。

19 イタリック補正と `\xkanjiskip`

2017/10/28 に導入済み。

19.1 直前の JFM 由来スペースの削除【コミュニティ版独自】

2018/03/09 に導入済み。

20 `tabular` 環境のセル内の JFM グルー

2018/03/09 に導入済み。

21 PDF のブックマークとアクセント文字

2018/07/28 に導入済み。