

upL^AT_EX 2_ε について

Ken Nakano & Japanese T_EX Development Community & TTK

作成日：2016/06/06

注意：

これは、アスキーのオリジナル版から fork したコミュニティ版 pL^AT_EX 2_ε の付属文書を upL^AT_EX 2_ε 用に書き換えたものです。

アスキー pT_EX には、高品質の日本語組版ソフトウェアとしてデファクトスタンダードの地位にあるといえます。しかし、(1) 直接使える文字集合が原則的に JIS X 0208 (JIS 第 1,2 水準) の範囲に限定されていること、(2) 8bit の非英語欧文との親和性が高いとは言えないこと、(3) pT_EX の利用が日本語に限られ、中国語・韓国語との混植への利用が進んでいないこと、といった弱点がありました。

これらの弱点を克服するため、pT_EX の内部コードを Unicode 化した拡張版が upT_EX です。また、upT_EX 上で用いる Unicode 版 pL^AT_EX が upL^AT_EX です¹。現在の upL^AT_EX は、日本語 T_EX 開発コミュニティが配布しているコミュニティ版 pL^AT_EX²をベースにしています。開発中の版は pL^AT_EX と同様に、GitHub のリポジトリ³で管理しています。

より詳細な upT_EX や upL^AT_EX の情報は、それぞれ README_{uplatex}.txt や README_{uptex}.txt などのテキストファイルを参照してください。

¹<http://www.t-lab.opal.ne.jp/tex/uptex.html>

²<https://github.com/texjporg/platex>

³<https://github.com/texjporg/uplatex>

1 概要

この文書は、 $\text{up}\text{\LaTeX} 2_{\epsilon}$ の概要を示していますが、使い方のガイドではありません。元となっている $\text{p}\text{\LaTeX} 2_{\epsilon}$ や $\text{\LaTeX} 2_{\epsilon}$ については、それぞれ $\text{p}\text{\LaTeX} 2_{\epsilon}$ と $\text{\LaTeX} 2_{\epsilon}$ の付属文書を参照してください。

この文書の構成は次のようになっています。

第 1 節 この節です。この文書についての概要を述べています。

第 2 節 $\text{up}\text{\LaTeX} 2_{\epsilon}$ で拡張した機能についての概要です。付属のクラスファイルやパッケージファイルについても簡単に説明しています。

付録 A この文書ソースの `DOCSTRIP` のためのオプションについて述べています。

付録 B $\text{up}\text{\LaTeX} 2_{\epsilon}$ の `dtx` ファイルをまとめて一つの `DVI` ファイルにするための文書ファイルの説明をしています。

付録 C 付録 B で説明をした文書ファイルを処理する `sh` スクリプト（手順）、`DOCSTRIP` 文書ファイル内の入れ子の対応を調べる `perl` スクリプトなどについて説明しています。

2 $\text{up}\text{\LaTeX} 2_{\epsilon}$ の機能について

$\text{up}\text{\LaTeX} 2_{\epsilon}$ の機能は、いくつかのファイルに分割されて実装されています。これらのファイルはつぎの 3 種類に分類することができます。

- フォーマットファイル
- クラスファイル
- パッケージファイル

フォーマットファイルには、基本的な機能が定義されており、 $\text{up}\text{\LaTeX} 2_{\epsilon}$ の核となるファイルです。このファイルに定義されているマクロは、実行時の速度を高めるために、あらかじめ \TeX の内部形式の形で保存されています。

クラスファイルは文書のレイアウトを設定するファイル、パッケージファイルはマクロの拡張を定義するファイルです。前者は `\documentclass` コマンドを用いて読み込み、後者は `\usepackage` コマンドを用いて読み込みます。

古い $\text{p}\text{\LaTeX} 2.09$ ユーザへの注意：

$\text{up}\text{\LaTeX}$ は新しいマクロパッケージですので、2.09 互換モードはサポートしていません。 $\text{\LaTeX} 2_{\epsilon}$ の仕様に従ってドキュメントを作成してください。

2.1 フォーマットファイル

フォーマットファイルには、基本的な機能が定義されていますが、これらは \TeX の内部形式に変換された形式となっています。フォーマットファイルを作成するには、ソースファイル “uplatex.ltx” を `iniuptex` プログラムで処理します。ただし、 \TeX Live や W32\TeX ではこの処理を簡単にする `fmtutil` あるいは `fmtutil-sys` というプログラムが用意されています。以下を実行すれば、フォーマットファイル `uplatex.fmt` が作成されます。

```
fmtutil --byfmt uplatex
```

次のリストが、“uplatex.ltx” の内容です。ただし、このバージョンでは、 \LaTeX から $\text{up\LaTeX 2}_{\epsilon}$ への拡張を `uplcore.ltx` をロードすることで行ない、`latex.ltx` には直接、手を加えないようにしています。したがって `uplatex.ltx` はとても短いものとなっています。`latex.ltx` には \LaTeX のコマンドが、`uplcore.ltx` には $\text{up\LaTeX 2}_{\epsilon}$ で拡張したコマンドが定義されています。

```
1 <plcore>
2 \let\orgdump\dump
3 \let\dump\relax
4 \input latex.ltx
5 \edef\platexBANNER{\the\everyjob}% save LaTeX banner
6 \typeout{*****^J%
7         *^J%
8         * making upLaTeX format^^J%
9         *^J%
10        *****}
11 \makeatletter
12 \input uplcore.ltx
13 \the\everyjob
14 \let\dump\orgdump
15 \let\orgdump\@undefined
16 \makeatother
17 \dump
18 <plcore>\endinput
19 </plcore>
```

実際に $\text{up\LaTeX 2}_{\epsilon}$ への拡張を行なっている `uplcore.ltx` は、`DOCSTRIP` プログラムによって、次のファイルの断片が連結されたものです。

- `uplvers.dtx` は、 $\text{up\LaTeX 2}_{\epsilon}$ のフォーマットバージョンを定義しています。
- `uplfonts.dtx` は、`NFSS2` を拡張しています。
- このほか、 $\text{p\LaTeX 2}_{\epsilon}$ に含まれる `plcore.dtx` をそのまま利用しています。これは、上記以外のコマンドでフォーマットファイルに格納されるコマンドを定義しています。

プリロードフォントや組版パラメータなどの設定は、`upldefs.ltx` をロードすることで行なっています。このファイルに記述されている設定を変更すれば、`upLATEX 2ε` をカスタマイズすることができます。カスタマイズする場合は、このファイルを直接、修正するのではなく、`upldefs.cfg` という名前でコピーをして、そのファイルを編集します。`upldefs.cfg` は `upldefs.ltx` の代わりに読み込まれます。

2.1.1 バージョン

`upLATEX 2ε` のバージョンやフォーマットファイル名は、`uplvers.dtx` で定義しています。これは、`pLATEX 2ε` のバージョンやフォーマットファイル名が `plvers.dtx` で定義されているのと同じです。

2.1.2 NFSS2 コマンド

`LATEX` では、フォント選択機構として `NFSS2` を用いています。`pLATEX 2ε` では、オリジナルの `NFSS2` と同様のインターフェイスで、和文フォントを選択できるように、`plfonts.dtx` で `NFSS2` を拡張していますので、`upLATEX 2ε` も `uplfonts.dtx` で同じ方式を採用しています。

`uplfonts.dtx` ファイルでは、`NFSS2` コマンドの定義のほか、プリロードフォントの設定、和文エンコードの定義、組版パラメータなどの設定、フォント定義ファイルなどの記述も含まれています。

2.2 クラスファイルとパッケージファイル

`upLATEX 2ε` が提供をする、クラスファイルやパッケージファイルのいくつかは、`pLATEX 2ε` に含まれるファイルを修正しています。

`upLATEX 2ε` に付属のクラスファイルは、次のとおりです。

- `ujbook.cls`, `ujarticle.cls`, `ujreport.cls`
横組用の標準クラスファイル。`ujclasses.dtx` から作成される。
- `utbook.cls`, `utarticle.cls`, `utreport.cls`
縦組用の標準クラスファイル。`ujclasses.dtx` から作成される。

また、`upLATEX 2ε` に付属のパッケージファイルは、次のとおりです。

- `uptrace.sty`
`LATEX` でフォント選択コマンドのトレースに使う `tracefnt.sty` が再定義してしまう `NFSS2` コマンドを、`upLATEX 2ε` 用に再々定義するためのパッケージ。`uplfonts.dtx` から作成される。

3 旧バージョンとの互換性

ここでは、このバージョンと以前のバージョンとの互換性や拡張部分について説明をしています。

3.1 p \LaTeX との互換性

up \LaTeX 2 ϵ は、p \LaTeX の上位互換という形を取っていますので、クラスファイルやいくつかのコマンドを置き換えるだけで、たいていの p \LaTeX 文書を簡単に up \LaTeX 文書に変更することができます。ただし、up \LaTeX では p \LaTeX で問題が指摘されていたフォントメトリックの不都合などいくつかのパラメータを変更していますので、レイアウトが変化することがあります。

また、up \LaTeX は新しいマクロパッケージですので、2.09 互換モードをサポートしていません。L \TeX 2 ϵ の仕様に従ってドキュメントを作成してください。

p \LaTeX 向けあるいは L \TeX 向けに作られた多くのクラスファイルやパッケージファイルはそのまま使えると思います。ただし、p \LaTeX 標準の漢字エンコーディング (JY1, JT1) を前提としたものが up \LaTeX で採用した漢字エンコーディング (JY2, JT2) と合致しないといったエラーが発生することもあります。用いようとしているクラスファイルやパッケージファイルがうまく動くかどうかを、完全に確かめる方法は残念ながらありません。一番簡単なのは、動かしてみることです。不幸にもうまく動かない場合は、ログファイルや付属の文書ファイルを参考に原因を調べてください。

3.2 latexrelease パッケージへの対応

L \TeX <2015/01/01>で導入された latexrelease パッケージをもとに、新しい p \LaTeX では platexrelease パッケージが用意されました。本来は up \LaTeX でも同様のパッケージを用意するのがよいのですが、現在は p \LaTeX から up \LaTeX への変更点が含まれていませんので、幸い platexrelease パッケージをそのまま用いることができます。このため、up \LaTeX で独自のパッケージを用意することはしていません。platexrelease パッケージを用いると、過去の up \LaTeX をエミュレートしたり、フォーマットを作り直すことなく新しい up \LaTeX を試したりすることができます。詳細は platexrelease のドキュメントを参照してください。

A DOCSTRIP プログラムのためのオプション

この文書のソース (uplatex.dtx) を DOCSTRIP プログラムによって処理することによって、いくつかの異なるファイルを生成することができます。DOCSTRIP プログ

ラムの詳細は、`docstrip.dtx` を参照してください。

この文書の DOCSTRIP プログラムのためのオプションは、次のとおりです。

| オプション | 意味 |
|--------|--|
| plcore | フォーマットファイルを作るためのファイルを生成 |
| pldoc | upL ^A T _E X 2 _ε のソースファイルをまとめて組版するための文書ファイルを生成 |
| shprog | 上記のファイルを作成するための sh スクリプトを生成 |
| plprog | 入れ子構造を調べる簡単な perl スクリプトを生成 |
| Xins | 上記の sh スクリプトや perl スクリプトを取り出すための DOCSTRIP バッチファイルを生成 |

A.1 ファイルの取り出し方

たとえば、この文書の “plcore” の部分を “`uplatex.ltx`” というファイルにする時の手順はつぎのようになります。

1. `uplatex docstrip`
2. 入力ファイルの拡張子 (`dtx`) を入力する。
3. 出力ファイルの拡張子 (`ltx`) を入力する。
4. DOCSTRIP オプション (`plcore`) を入力する。
5. 入力ファイル名 (`uplatex`) を入力する。
6. `uplatex.ltx` が存在する場合は、確認を求めてくるので、“y” を入力する。
7. 別の処理を行なうかを問われるので、“n” を入力する。

これで、`uplatex.ltx` が作られます。

あるいは、次のような内容のファイル `fmt.ins` を作成し、`uplatex fmt.ins` することでも `uplatex.ltx` を作ることができます。

```
\def\batchfile{fmt.ins}
\input docstrip.tex
\generateFile{uplatex.ltx}{t}{\from{uplatex.dtx}{plcore}}
```

B 文書ファイル

ここでは、このパッケージに含まれている dtx ファイルをまとめて組版をするための文書ファイルについて説明をしています。個別に処理した場合と異なり、変更履歴や索引も付きます。全体で、およそ 120 ページ程度になります。

`filecontents` 環境は、引数に指定されたファイルが存在するときは何もませんが、存在しないときは、環境内の内容でファイルを作成します。`upldoc.dic` ファイルは、`mendex` プログラムで索引を処理するときに `\西暦`、`\和暦` に対する「読み」を付けるために必要です。

```
20 <{*pldoc}
21 \begin{filecontents}{upldoc.dic}
22 西暦      せいれき
23 和暦      われき
24 \end{filecontents}
```

文書クラスには、`jltxdoc` クラスを用います。

```
25 \documentclass{jltxdoc}
26 %\usepackage{plex} %% comment out for upLaTeX
27 \listfiles
28
```

いくつかの \TeX プリミティブとコマンドを索引に出力しないようにします。

```
29 \DoNotIndex{\def,\long,\edef,\xdef,\gdef,\let,\global}
30 \DoNotIndex{\if,\ifnum,\ifdim,\ifcat,\ifmmode,\ifvmode,\ifhmode,%
31             \iftrue,\iffalse,\ifvoid,\ifx,\ifeof,\ifcase,\else,\or,\fi}
32 \DoNotIndex{\box,\copy,\setbox,\unvbox,\unhbox,\hbox,%
33             \vbox,\vtop,\vcenter}
34 \DoNotIndex{\@empty,\immediate,\write}
35 \DoNotIndex{\egroup,\bgroup,\expandafter,\begingroup,\endgroup}
36 \DoNotIndex{\divide,\advance,\multiply,\count,\dimen}
37 \DoNotIndex{\relax,\space,\string}
38 \DoNotIndex{\csname,\endcsname,\@spaces,\openin,\openout,%
39             \closein,\closeout}
40 \DoNotIndex{\catcode,\endinput}
41 \DoNotIndex{\jobname,\message,\read,\the,\m@ne,\noexpand}
42 \DoNotIndex{\hsize,\vsize,\hskip,\vskip,\kern,\hfil,\hfill,\hss,\vss,\unskip}
43 \DoNotIndex{\m@ne,\z@,\z@skip,\@ne,\tw@,\p@,\@minus,\@plus}
44 \DoNotIndex{\dp,\wd,\ht,\setlength,\addtolength}
45 \DoNotIndex{\newcommand,\renewcommand}
46
```

索引と変更履歴の見出しに `\part` を用いるように設定をします。

```
47 \IndexPrologue{\part*{索引}}%
48             \markboth{索引}{索引}%
49             \addcontentsline{toc}{part}{索引}}%
50 イタリアック体の数字は、その項目が説明されているページを示しています。
51 下線の引かれた数字は、定義されているページを示しています。
```

52 その他の数字は、その項目が使われているページを示しています。}

53 %

54 \GlossaryPrologue{\part*{変更履歴}}%

55 \markboth{変更履歴}{変更履歴}%

56 \addcontentsline{toc}{part}{変更履歴}}

57

標準の\changes コマンドを、複数ファイルの文書に合うように修正しています。

58 \makeatletter

59 \def\changes@#1#2#3{%

60 \let\protect\@unexpandable\protect

61 \edef\@tempa{\noexpand\glossary{#2\space\currentfile\space#1\levelchar

62 \ifx\saved@macroname\@empty

63 \space\actualchar\generalname

64 \else

65 \expandafter\@gobble

66 \saved@macroname\actualchar

67 \string\verb\quotechar*%

68 \verbatimchar\saved@macroname

69 \verbatimchar

70 \fi

71 :\levelchar #3}}%

72 \@tempa\endgroup\@esphack}

73 \makeatother

74 \RecordChanges

75 \CodelineIndex

76 \EnableCrossrefs

77 \setcounter{IndexColumns}{2}

78 \settowidth\MacroIndent{\ttfamily\scriptsize 000\ }

ここからが本文ページとなります。

79 \begin{document}

80 \title{The up\LaTeXe\ Sources}

81 \author{Ken Nakano \& Japanese \TeX\ Development Community \& TTK}

82

83 % This command will be used to input the patch file

84 % if that file exists.

85 \newcommand{\includeltpatch}{%

86 \def\currentfile{uplpatch.ltx}

87 \part{uplpatch}

88 {\let\ttfamily\relax

89 \xdef\filekey{\filekey, \thepart={\ttfamily\currentfile}}}%

90 Things we did wrong\ldots

91 \IndexInput{uplpatch.ltx}}

92

93 % Get the date from uplvers.dtx

94 \makeatletter

95 \def\patchdate{0}%% Modified (May 8, 2016)

96 \begingroup

97 \def\ProvidesFile#1\pfmtversion#2{\date{#2}\endinput}


```

98   \input{uplvers.dtx}
99   \global\let\X@date=\@date
100
101  % Add the patch version if available.
102   \long\def\Xdef#1#2#3\def#4#5{%
103     \xdef\X@date{#2}%
104     \xdef\patchdate{#5}%
105     \endinput}%
106   \InputIfFileExists{uplpatch.ltx}
107     {\let\def\Xdef}{\global\let\includeltpatch\relax}
108 \endgroup
109
110 \ifx\@date\X@date
111   \def\Xpatch{0}
112   \ifx\patchdate\Xpatch\else
113     \edef\@date{\@date\space Patch level\patchdate}
114   \fi
115 \else
116   \@warning{uplpatch.ltx does not match uplvers.dtx!}
117   \let\includeltpatch\relax
118 \fi
119 \makeatother
120
121 \pagenumbering{roman}
122 \maketitle
123 \renewcommand\maketitle{}
124 \tableofcontents
125 \clearpage
126 \pagenumbering{arabic}
127
128 \DocInclude{uplvers}    % upLaTeX version
129
130 \DocInclude{uplfonts}  % NFSS2 commands
131
132 %\DocInclude{plcore}    % kernel commands (comment out for upLaTeX)
133
134 %\DocInclude{plext}     % external commands (comment out for upLaTeX)
135
136 %\DocInclude{pl209}     % 2.09 compatibility mode commands (comment out for upLaTeX)
137
138 \DocInclude{ukinsoku}   % kinsoku parameter
139
140 \DocInclude{ujclasses}  % Standard class
141
142 %\DocInclude{jltxdoc}   % dtx documents class (comment out for upLaTeX)
143
144 %\includeltpatch        % patch file (comment out May 8, 2016)
145

```

ltxdoc.cfg に \AtEndOfClass{\OnlyDescription} が指定されている場合は、こ

ここで終了します。

```
146 \StopEventually{\end{document}}
147
```

変更履歴と索引を組版します。変更履歴ファイルと索引の作り方の詳細については、おまけ C.1 を参照してください。

```
148 \clearpage
149 \pagestyle{headings}
150 % Make TeX shut up.
151 \hbadness=10000
152 \newcount\hbadness
153 \hfuzz=\maxdimen
154 %
155 \PrintChanges
156 \clearpage
157 %
158 \begingroup
159   \def\endash{--}
160   \catcode'\- \active
161   \def-\{\futurelet\temp\indexdash}
162   \def\indexdash{\ifx\temp-\endash\fi}
163
164   \PrintIndex
165 \endgroup
```

ltxdoc.cfg に 2 度目の \PrintIndex が指定されているかもしれません。そこで、最後に、変更履歴や索引が 2 度組版されないように \PrintChanges および \PrintIndex コマンドを何も実行しないようにします。

```
166 \let\PrintChanges\relax
167 \let\PrintIndex\relax
168 \end{document}
169 \end{preamble}
```

C おまけプログラム

C.1 シェルスクリプト mkpldoc.sh

up⁴LaTeX 2_ε のマクロ定義ファイルをまとめて組版するときに便利なシェルスクリプトです。このシェルスクリプト⁴の使用方法は次のとおりです。

```
sh mkpldoc.sh
```

⁴このシェルスクリプトは UNIX 用です。しかし rm コマンドを delete コマンドにするなどすれば、簡単に DOS などのバッチファイルに修正することができます。

C.1.1 mkpldoc.sh の内容

まず、以前に upldoc.tex を処理したときに作成された、目次ファイルや索引ファイルなどを削除します。

```
170 (*shprog)
171 for f in upldoc.toc upldoc.idx upldoc.glo ; do
172 if [ -e $f ]; then rm $f; fi
173 done
```

そして、ltxdoc.cfg を空にします。このファイルは、jltxdoc.cls の定義を変更するものですが、ここでは、変更されたくありません。

```
174 echo "" > ltxdoc.cfg
```

そして、upldoc.tex を処理します。

```
175 uplatex upldoc.tex
```

索引と変更履歴を作成します。このスクリプトでは、変更履歴や索引を生成するのに mendex プログラムを用いています。mendex は makeindex の上位互換のファイル整形コマンドで、索引語の読みを自動的に付けるなどの機能があります。

upL^AT_EX の場合は mendex コマンドを UTF-8 モードで実行する必要がありますので、-U というオプションを付けます⁵。makeindex コマンドには、このオプションがありません。

-s オプションは、索引ファイルを整形するためのスタイルオプションです。索引用の gind.ist と変更履歴用の gglo.ist は、L^AT_EX のディストリビューションに付属しています。

-o は、出力するファイル名を指定するオプションです。

-f は、項目に“読み”がなくてもエラーとしないオプションです。makeindex コマンドには、このオプションがありません。

```
176 mendex -U -s gind.ist -d upldoc.dic -o upldoc.ind upldoc.idx
177 mendex -U -f -s gglo.ist -o upldoc.gls upldoc.glo
```

ltxdoc.cfg の内容を\includeonly{}にし、upldoc.tex を処理します。このコマンドは、引数に指定されたファイルだけを“\include”するためのコマンドですが、ここでは何も\include したくないので、引数には何も指定をしません。しかし、\input で指定されているファイルは読み込まれます。したがって、目次や索引や変更履歴のファイルが処理されます。この処理は、主に、これらでエラーが出るかどうかの確認です。

```
178 echo "\includeonly{}" > ltxdoc.cfg
179 uplatex upldoc.tex
```

⁵uplatex コマンドも実際には UTF-8 モードで実行する必要がありますが、デフォルトの内部漢字コードが UTF-8 に設定されているはずですので、-kanji=utf8 を付けなくても処理できると思います。

最後に、再び `ltxdoc.cfg` を空にして、`upldoc.tex` を処理をします。本文を 1 ページから開始していますので、この後、もう一度処理をする必要はありません。

```
180 echo "" > ltxdoc.cfg
181 uplatex upldoc.tex
182 # EOT
183 </shprog>
```

C.2 perl スクリプト `dstcheck.pl`

DOCSTRIP 文書ファイルは、 \LaTeX のソースとその文書を同時に管理する方法として、とてもすぐれていると思います。しかし、たとえば `jclasses.dtx` のように、条件が多くなると、入れ子構造がわからなくなってしまうがちです。 \LaTeX で処理すれば、エラーによってわかりますが、文書ファイルが大きくなると面倒です。

ここでは、DOCSTRIP 文書ファイルの入れ子構造を調べるのに便利な、perl スクリプトについて説明をしています。

この perl スクリプトの使用方法是次のとおりです。

```
perl dstcheck.pl file-name
```

C.2.1 `dstcheck.pl` の内容

最初に、この perl スクリプトが何をするのかを簡単に記述したコメントを付けます。

```
184 <*plprog>
185 ##
186 ## DOCSTRIP 文書内の環境や条件の入れ子を調べる perl スクリプト
187 ##
```

このスクリプトは、入れ子の対応を調べるために、次のスタックを用います。〈条件〉あるいは〈環境〉を開始するコードが現れたときに、それらはスタックにプッシュされ、終了するコードでポップされます。したがって、現在の〈条件〉あるいは〈環境〉と、スタックから取り出した〈条件〉あるいは〈環境〉と一致すれば、対応が取れているといえます。そうでなければエラーです。

@dst スタックには、〈条件〉が入ります。条件の開始は、“%<*(条件)>” です。条件の終了は、“%</(条件)>” です。〈条件〉には、>文字が含まれません。@env スタックには、〈環境〉が入ります。

先頭を明示的に示すために、ダミーの値を初期値として用います。スタックは、〈条件〉あるいは〈環境〉の名前と、その行番号をペアにして操作をします。

```
188 push(@dst,"DUMMY"); push(@dst,"000");
189 push(@env,"DUMMY"); push(@env,"000");
```

このwhile ループの中のスクリプトは、文書ファイルの 1 行ごとに実行をします。

```
190 while (<>) {
```

入力行が条件を開始する行なのかを調べます。条件の開始行ならば、@dst スタックに〈条件〉と行番号をプッシュします。

```
191 if (/^%<\*([>]+)>/) { # check conditions
192     push(@dst,$1);
193     push(@dst,$.);
```

そうでなければ、条件の終了行なのかを調べます。現在行が条件の終了を示している場合は、@dst スタックをポップします。

```
194 } elsif (/^%<\*([>]+)>/) {
195     $linenum = pop(@dst);
196     $conditions = pop(@dst);
```

現在行の〈条件〉と、スタックから取り出した〈条件〉が一致しない場合、その旨のメッセージを出力します。

なお、DUMMY と一致した場合は、一番外側のループが合っていないということを示しています。このとき、これらのダミー値をスタックに戻します。いつでもスタックの先頭をダミー値にするためです。

```
197 if ($1 ne $conditions) {
198     if ($conditions eq "DUMMY") {
199         print "ARGV: '</$1>' (1.$.) is not started.\n";
200         push(@dst,"DUMMY");
201         push(@dst,"000");
202     } else {
203         print "ARGV: '<*$conditions>' (1.$linenum) is ended ";
204         print "by '<*$1>' (1.$.)\n";
205     }
206 }
207 }
```

環境の入れ子も条件と同じように調べます。

verbatim 環境のときに、その内側をスキップしていることに注意をしてください。

```
208 if (/^% *\begin\{verbatim\}/) { # check environments
209     while(<>) {
210         last if (/^% *\end\{verbatim\}/);
211     }
212 } elsif (/^% *\begin\{([~}]+)\}\{(.*)\}/) {
213     push(@env,$1);
214     push(@env,$.);
215 } elsif (/^% *\begin\{([~}]+)\}\{([~}]+)\}/) {
216     push(@env,$1);
217     push(@env,$.);
218 } elsif (/^% *\end\{([~}]+)\}\{([~}]+)\}/) {
219     $linenum = pop(@env);
220     $environment = pop(@env);
221     if ($1 ne $environment) {
222         if ($environment eq "DUMMY") {
223             print "ARGV: '\\end{$1}' (1.$.) is not started.\n";
```

```

224     push(@env, "DUMMY");
225     push(@env, "000");
226   } else {
227     print "$ARGV: \\begin{$environment} (l.$linenum) is ended ";
228     print "by \\end{$1} (l.$.)\n";
229   }
230 }
231 }

```

ここまでが、最初のwhile ループです。

```
232 }
```

文書ファイルを読み込んだ後、終了していない条件があるかどうかを確認します。すべての条件の対応がとれていれば、この時点での@dst スタックにはダミー値しか入っていません。したがって、対応が取れている場合は、最初の2つのポップによって、ダミー値が設定されます。ダミー値でなければ、ダミー値になるまで、取り出した値を出力します。

```

233 $linenum = pop(@dst);
234 $conditions = pop(@dst);
235 while ($conditions ne "DUMMY") {
236   print "$ARGV: '<*$conditions>' (l.$linenum) is not ended.\n";
237   $linenum = pop(@dst);
238   $conditions = pop(@dst);
239 }

```

環境の入れ子についても、条件の入れ子と同様に確認をします。

```

240 $linenum = pop(@env);
241 $environment = pop(@env);
242 while ($environment ne "DUMMY") {
243   print "$ARGV: '\\begin{$environment}' (l.$linenum) is not ended.\n";
244   $linenum = pop(@env);
245   $environment = pop(@env);
246 }
247 exit;
248 </plprog>

```

C.3 DOCSTRIP バッチファイル

ここでは、付録 C.1 と付録 C.2 で説明をした二つのスクリプトを、このファイルから取り出すための DOCSTRIP バッチファイルについて説明をしています。

まず、DOCSTRIP パッケージをロードします。また、実行経過のメッセージを出力しないようにしています。

```

249 <*Xins>
250 \input docstrip
251 \keepsilent

```

DOCSTRIP プログラムは、連続する二つのパーセント記号 (%%) ではじまる行をメタコメントとみなし、条件によらず出力をします。しかし、“%” は \TeX ではコメントであっても、sh や perl にとってはコメントではありません。そこで、メタコメントとして出力する文字を “##” と変更します。

```
252 {\catcode'\#12 \gdef\MetaPrefix{## }}
```

そして、プリアンブルに出力されるメッセージを宣言します。ここでは、とくに何も指定していませんが、宣言をしないとデフォルトの記述が ‘%%’ 付きで出力されてしまうため、それを抑制する目的で使っています。

```
253 \declarepreamble\thispre
```

```
254 \endpreamble
```

```
255 \usepreamble\thispre
```

ポストアンブルも同様に、宣言をしないと ‘\endinput’ が出力されます。

```
256 \declarepostamble\thispost
```

```
257 \endpostamble
```

```
258 \usepostamble\thispost
```

`\generate` コマンドで、どのファイルに、どのファイルのどの部分を出力するのかを指定します。

```
259 \generate{
```

```
260   \file{dstcheck.pl}{\from{uplatex.dtx}{plprog}}
```

```
261   \file{mkpldoc.sh}{\from{uplatex.dtx}{shprog}}
```

```
262 }
```

```
263 \endbatchfile
```

```
264 \Xins
```

変更履歴

| | | | |
|--|---|--------------------------------------|---|
| 2011/05/07 v1.0c-u00 | | を除外 | 8 |
| ・ p \LaTeX 用から up \LaTeX 用に修正。 | 1 | 2016/05/12 v1.0i-u00 | |
| 2016/04/06 v1.0e-u00 | | ・ 一時コマンド <code>\orgdump</code> を最終的 | |
| ・ p \LaTeX の変更に従。 | 1 | に未定義へ | 3 |
| 2016/05/07 v1.0g-u00 | | 2016/05/21 v1.0k-u00 | |
| ・ フォーマット作成時に \LaTeX のバ | | ・ p \LaTeX の変更に従。 | 1 |
| ナーを一旦保存 | 3 | 2016/06/06 v1.0k-u01 | |
| 2016/05/08 v1.0h-u00 | | ・ up \LaTeX 用にドキュメントを全体 | |
| ・ ドキュメントから <code>uplpatch.ltx</code> | | 的に改訂 | 1 |