

日本語 BibT_EX : jBibT_EX

松井 正一 (matsui@denken.or.jp)

(財) 電力中央研究所 情報システム部 (〒100 千代田区大手町 1-6-1)

1991 年 1 月 1 日

概要

jBibT_EX は BibT_EX 0.99c を基に bib, bst ファイル中に漢字コードをいれるように変更を加えたものである。

このメモでは、先ず最初に BibT_EX の概要を簡単に説明した後、日本語化に対する基本的な考え方、主な変更点、日本語に対応するように作成したスタイルについて説明する。

1 はじめに

T_EX を NTT の齊藤氏が日本語化してくれたおかげで T_EX で日本語が使えるようになった¹。また NTT の磯崎氏が L^AT_EX の日本語版 jL^AT_EX を作ってくれたので、簡単に日本語のタイプセットができるようになった²。

技術系の文書を書く場合には、参考文献の管理に手間取ることが多い。この手間を軽減するためのプログラムとして L^AT_EX には、roff 一族の refer に対応した BibT_EX があるが、この日本語化ができていなかった³、jL^AT_EX の機能を活かしきれないくらいがあった⁴。そこで、とりあえず日本語が使える形の BibT_EX を jBibT_EX という形で作成してみた。

2 BibT_EX の概要

2.1 処理の流れ

BibT_EX は L^AT_EX で作成される aux ファイル⁵ 中の \citation に示されている \cite 項目に対応する参考文献レコードを L^AT_EX の文書中に \bibliography

¹ ICOT の高木氏 (現在日立)、都立大の桜井氏の力に負うところも大きい。

² 最初に開発した jBibT_EX は NTT 版を想定していたので、ASCII 版に対する記述がないだけで、他意はない。現在は NTT 版、ASCII 版の両方を使っている。

³ 1988 年 4 月に、高木氏から jL^AT_EX in C のテープを入手した時点で、また ASCII バージョンを知らなかったのだ。

⁴ 私は面倒くさいことは嫌いなので。

⁵ ファイルのエクステンションでファイル名を代表する。

で指定されたデータベースファイル (`bib`) から検索し, `\bibliographystyle` で指定された参考文献書式 (スタイル) で `bbl` ファイルに書き出す形で動作する. 実行時のメッセージは端末に出力されると同時に, ログファイルの `blg` ファイルにも出力される.

通常はある文書 `file.tex` を処理するには次のような手順をふむ⁶.

1. `latex file` で `aux` を作り出す.
2. `bibtex file` で `aux`, `bib` から `bbl` ファイルを作り出す.
3. 以後の `latex file` の起動で `bbl` が自動的に読み込まれる.

2.2 参考文献データベース

データベース中の参考文献レコードは, 文献の形式 (論文, 本など) を指定した後に, 引用キー, 書誌項目を “キーワード=値” の形でカンマで区切って並べる形で記述する [1, 2].

⁶ `bbl` ファイルの内容が固定された後は, `latex` を起動するだけでよい (相互参照関係の処理のために 2 回以上の起動が必要なこともある).

文献の形式としては以下が用意されている。

文献の形式	説明
article	論文誌など発表された論文
book	出版社の明示された本
booklet	印刷，製本されているが出版主体が不明なもの
conference	inproceedings と同じ (Scribe との互換性のため)
inbook	書物の一部 (章，節，文など何でも)
incollection	それ自身の表題を持つ，本の一部分
inproceedings	会議録中の論文
manual	マニュアル
masterthesis	修士論文
misc	他のどれにも当てはまらない時に使う
phdthesis	博士論文
proceedings	会議録
techreport	テクニカルレポート
unpublished	正式には出版されていないもの

それぞれで必須な項目 (フィールド) がある。

項目 (フィールド) としては以下がある⁷。

⁷ 正確には plain, alpha, abbrev, unsrt で, annotate を除いた項目が意味を持つ (タイプセットに使われる)。

フィールド名	説 明
address	出版主体の住所
annote	注釈付きのスタイルで使われる
author	著者名
booktitle	本の名前
chapter	章，節などの番号
crossref	相互参照する文献のデータベースのキー
edition	本の版
editor	編集者
howpublished	どのようにしてこの奇妙なものが発行されたか
institution	テクニカルレポートの発行主体
journal	論文誌名
key	著者名がない時に相互引用，ラベル作成などに使われる
month	発行月，または書かれた月
note	読者に役立つ付加情報
number	論文誌などの番号
organization	会議を主催した機関名あるいはマニュアルの出版主体
pages	ページ (範囲)
publisher	主版社 (者) 名
school	修士，博士論文が書かれた大学名
series	シリーズ，あるいは複数巻の本の名前
title	表題
type	テクニカルレポートの種類
volume	論文誌，複数巻の本の巻
year	発行年，または書かれた年

以下にデータベースの記述例を示す.

```
@ARTICLE{la,
  author="Masao Iri and Kazuo Murota and Shouichi Matsui",
  title="Linear-time approximation algorithms for finding the
        minimum-weight perfect matching on a plane",
  journal=ipl,volume=12,pages="206--209",year=1981
}
```

頻繁に使われる論文誌名などは，(bst ファイル中に) 後述のマクロ機能を使って省略形を定義しておくことで，journal=ipl などと書ける (bib ファイル中に@STRING(ipsj={ 情報処理学会 }) のように定義することもできる).

2.3 スタイル

スタイルは `alpha.bst` などのスタイルファイルに記述されている。スタイルファイル中の書式は、スタックマシンに対しての命令を後置演算子記法で記述した形である [3]。

スタイルを記述する言語には二種類の関数がある。一つはシステム組み込みのものであり、`newline$`、`write$` などのように名前の最後が '\$' で終る。もう一つは `MACRO`、`FUNCTION` を使って定義するものである。

組み込み関数には以下のようなものがある (全部で 37 ある)。

組み込み関数 (名)	機能
<code>></code> , <code><</code> , <code>=</code>	整数の比較
<code>+</code> , <code>-</code>	整数加算, 減算
<code>*</code>	文字列の連結
<code>:=</code>	代入
<code>add.period\$</code>	文字列の ' <code>}</code> ' でない最後の文字が, ' <code>.</code> ', ' <code>?</code> ', ' <code>!</code> ' のいずれでもなければ ' <code>.</code> ' を最後に付加する
<code>call.type\$</code>	エントリの型 (文献の型) に応じた関数を呼び出す。例えば <code>book</code> なら関数 <code>book</code> を呼び出す。
<code>change.case\$</code>	指定に応じて大文字, 小文字に変換した文字列を返す (指定が ' <code>t</code> ' なら最初の文字以外をすべて小文字に, ' <code>l</code> ' ならすべて小文字に, ' <code>u</code> ' ならすべて大文字に変換する)。
<code>chr.to.int\$</code>	1 文字のみの文字列の ASCII コードを返す
<code>text.prefix\$</code>	高々指定された文字数以内の先頭文字列を返す
<code>skip\$</code>	何もしない
<code>substring\$</code>	部分文字列を返す
<code>write\$</code>	文字列を <code>bb1</code> ファイルに書き出す

スタイルは前述の関数とコマンドおよびシステム変数を使って記述する。コマンドは以下からなる。

- 宣言と変数, 関数定義のための 5 つのコマンド (`ENTRY`, `FUNCTION`, `INTEGERS`, `MACRO`, `STRINGS`)
- データベース読み込みのためのコマンド (`READ`)。データベースを読み込み, 内部的にリストの形式にする。

コマンド名	機 能
ENTRY	フィールドとエントリを宣言する
FUNCTION	新たな関数を定義する
INTEGERS	グローバルな整数型の変数を宣言する
MACROS	文字列マクロを定義する
STRINGS	グローバルな文字列型の変数を宣言する

- 処理と出力のための 4 つのコマンド (EXECUTE, ITERATE, REVERSE, SORT)

コマンド名	機 能
EXECUTE	1 つの関数を実行する
ITERATE	現在のリスト中の各エントリに対して 1 つの関数を適用する
REVERSE	ITERATE と同じ操作をリストの逆順に行う
SORT	sort.key\$ によってリストをソートする

システム変数には以下がある。

変数名	説 明
sort.key\$	文字列型, ソートのキー
entry.max\$	整数型, エントリ文字列の最大長 (100)
global.max\$	整数型, 文字列の最大長 (1000)

以下にスタイルファイル alpha の一部分を例として示す。

3 日本語化の概要

3.1 バージョン 0.10 と 0.20 の生い立ち

最小限の変更ですませることにし⁸, 以下の形で元の BiB_TE_X(Version 0.99c; Pascal 版) に変更を加えた第 1 バージョン (0.10) を 1988 年 5 月に作成した. このために行った変更は, change file の変更が 600 行程度, C で書かれたサポートルーチン (bibext.c) への変更が 50 行程度であるが, change file への本質的な変更は 200 行程度である.

その後 1989 年 5 月に BiB_TE_X の C バージョンに対応して, Pascal コンパイラーのないシステムでも動作するように, 第 2 バージョン (0.20) を作成した. このために行った変更は, 新たな組み込み関数を追加することも含めて

⁸ 手抜きである.

```

MACRO {tcs} {"Theoretical Computer Science"}
READ
FUNCTION {sortify}
  { purify$ "l" change.case$ }
FUNCTION {chop.word}
{ 's :=
  'len :=
  s #1 len substring$ =
    { s len #1 + global.max$ substring$ }
  's
  if$
}
INTEGERS { et.al.char.used }
FUNCTION {initialize.et.al.char.used}
  { #0 'et.al.char.used := }
EXECUTE {initialize.et.al.char.used}
ITERATE {presort}
SORT
EXECUTE {initialize.longest.label}
ITERATE {forward.pass}
REVERSE {reverse.pass}
FUNCTION {begin.bib}
{ et.al.char.used
  { "\newcommand{\etalchar}[1]{$^\{#1\}$}" write$ newline$ }
  'skip$
  if$
  preamble$ empty$
  'skip$
  { preamble$ write$ newline$ }
  if$
  "\begin{thebibliography}{\longest.label * "}" * write$ newline$
}
EXECUTE {begin.bib}
EXECUTE {init.state.consts}
ITERATE {call.type$}
FUNCTION {end.bib}
{ newline$
  "\end{thebibliography}" write$ newline$
}
EXECUTE {end.bib}

```

change file の変更が 450 行程度, bibext.c への変更が 430 行程度である⁹. bibext.c の変更の大部分は, \LaTeX 用に桜井氏が作成したプログラムを流用している¹⁰ので, 新たに作成した部分は少ない¹⁰.

⁹ web2c の出力中の putc を日本語コードに対応するための Fputc に sed するために Makefile の一部分も変更した.

¹⁰ 作業は SUN-3 で行ったが, その他のマシンへの移植作業は JUNET を通じて協力していた. これにより現在は, SUN-4, VAX/ Ultrix/BSD, NEWS /NEWS-OS, Symmetry /Dynix, FACOM M Series /UTS, LUNA /BSD, Σ WS など動作している.

3.2 バージョン 0.30 と MS-DOS 版の旅立ち

筆者が勤務の関係で米国出張中のクリスマス休暇の間に作成したのがバージョン 0.30 と MS-DOS 版の jBibTeX である¹¹。

バージョン 0.30 では ASCII 版の version 0.98 をベースにした Kanji BibTeX との整合性を向上させる、また不完全であった日本語対応を完全にするなどの作業を行った。このためにバージョン 0.20 から変更した change file の行数は 50 行程度である。これに際しては、職場の同僚である二方厚志、所健一の両氏、(株)アスキーの倉沢良一、渡邊雄一両氏の協力を得た。特に渡邊氏には多忙な中、様々なテストに協力していただいた。バージョン 0.30 ではドキュメント類の修正、スタイルファイルの改良も行った。これにより、NTT 版だけでなく、ASCII 版でも (ドキュメントの処理も含めて) 問題なく動作するバージョンの配布キットが完成した。

MS-DOS 版は UNIX 上で作成した C のソースを直接変更して Turbo-C を利用して作成した。コンパイラ、リンカの制約から大量の大域変数が宣言できないので、大域変数の宣言部分をいくつかのファイルに分けるとか、動的にメモリを割り当てるとかの変更を C ソースに直接施した。変更行数は 100 行程度である。利用可能メモリの制約から、処理可能引用文献数が UNIX 版に比べて少ないことを除けば (300 件程度以内でないとだめ)、実用上十分な性能を持つプログラムである。

3.3 漢字コードの扱い

オリジナルの BibTeX への変更を少なくするために、漢字コードは内部的には EUC で表現することとした。ファイルから一行を読み込む時点で、EUC ならそのまま、JIS 漢字コードは EUC にコード変換する¹²。

漢字コードをファイルに書き出す時にコード系の変換を行う。 jBibTeX は NTT 版 jTeX に改良を加えて漢字コードとして JIS の他に、EUC も使えるようにしたものの上の jATeX と対にして使うことを前提としているが、ASCII 版でも問題なく使える。この jTeX では漢字コードの指定が EUC/SJIS であっても、JIS 漢字コードも読み込めるように変更したものである¹³。

そこで第 1 バージョン (0.10) では漢字コードの変換は以下の形で行うこととした。

¹¹ 寒い所だったので、家族で外出する気にもならず、お金もなかったので旅行もできずで :-)。

¹² JIS 漢字コードのエスケープ・シーケンスの先頭の 2 バイトのみを漢字 in/out の判定に使用し、3 バイト目は何でもよいようにしてある。また、当所では SJIS は使っていないのでバージョン 0.10 ではそれは考えていなかったが、バージョン 0.20 ではそれにも対処できるようにした。MS-DOS 版では SJIS がデフォルトの漢字コードになっている。JIS も読める。

¹³ 都立大の桜井氏が中心となって開発しているバージョンに対応している。

- 補助 (aux), スタイル (bst), データベース (bib) ファイル中の漢字コードの総てが EUC 漢字コードであれば, bbl, blg ファイルあるいは端末への出力は EUC コードで行う.
- 上記以外の場合には, 書き出しが行れる直前に読んだ JIS 漢字のコードのエスケープ・シーケンスを使って漢字コードを書き出す¹⁴.

漢字コードの変換は, 入力時は bibext.c で行い, 出力時は bibtex 本体で行うこととしたために, change file の変更が多くなった.

第 2 バージョン (0.20) 以降では, 漢字コード系はサイト毎にコンパイル時に決定することとし, 利用者単位には次の環境変数で変更可能とした.

```

端末      BIBTERMCODE
ファイル BIBFILECODE

```

これらの変数に, j_EX の \kanjiterminaltype, \kanjifiletype で指定するのと同じ値を設定することで (EUC なら 20, SJIS なら 10 など) 入出力のコード系を変更できる (j_EX 同様 EUC, SJIS を指定してあっても JIS コード系も読み込めるようになっている).

漢字コードの変換はすべて bibext.c の中で行うこととしたので, change file の変更が少なくなっている.

3.4 組み込み関数の仕様変更

組み込み関数の substring\$, text.prefix\$ においては 1 つの漢字コードの間で (1 バイト目と 2 バイト目の間で) 文字列が切られないようにした. 例えば, ‘漢字’ の先頭から 3 文字 (バイト) を抜出すと ‘字’ の 2 バイト目がなくなってしまうので, このような場合には, 4 バイト目まで切出すようにした¹⁵.

前記との関係で, chr.to.int\$ は引数が漢字コードならば single character でなくても warning を出さず, 漢字文字列の 1 バイト目のコードを値として返すようにした¹⁶.

以上はバージョン 0.10 から行っていた変更であるが, バージョン 0.30 では, さらに次の 2 つの変更も追加した.

¹⁴ 完全に漢字コード対応になっていない可能性があるので, bib ファイルなどの記述にエラーがあった時に, 端末などに EUC コードのまま漢字コードが出力されるかもしれない (第 1 バージョン (0.10) の場合).

¹⁵ これは日本語用のスタイルでない alpha, plain などを使った場合でもそれなりの出力が得られるようにするためである.

¹⁶ これは 0.10 バージョンで試作したスタイルファイル中で, 漢字コード文字列の判定をいいかげんに行うために必要となった変更であり, 0.20 バージョンでは漢字コード文字列判定用に新たな組み込み関数を追加したので不要かも知れない.

`add.period$` 関数は、半角の ‘!’ ‘?’ ‘.’ で終われば、‘.’ を付加しないので、全角の ‘!’ ‘?’ ‘.’ ‘。’ で終わっても ‘.’ を付加しないようにした。

`format.name$` 関数では、漢字コードで表記された氏名の場合には全角の空白も半角の空白と同じように扱うこととした。また複数の名前は `and` を間に入れて列挙するが、漢字コード表記された氏名の場合には全角の “ ” “、 ” を使っても `and` と同じように扱うようにした。これで漢字コード著者名の入力が以前より簡単になるはずである (少なくとも PC でかな漢字変換 FEP を使っている場合には、半角への切り替えが不要になる)。

3.5 組み込み関数の追加

次に述べる日本語用のスタイルに関連することであるが、漢字コードを含む文字列の場合に処理を分ける必要がある。第 1 バージョンでは漢字コードを含むかどうかは、先頭文字のみで判定するようなスタイルを作成したが、これでは不完全であるので、第 2 バージョン以降からは `is.kanji.str$` という関数を追加した。この関数は、与えられた文字列が日本語コード (non ASCII コード) を含む時に “真” (整数の 1) を返す関数である。

4 日本語用のスタイル

4.1 問題点

前述の方法で日本語を扱えるようにしても，BibT_EX のスタイル (bst) を日本語に合せないと，参考文献が奇妙な型式となる．例えば，plain のスタイルを使うと，著者が複数の場合

[1] 松井 and 篠原 …

のような形になってしまう．日本語の場合にはやはり

[1] 松井, 篠原 …

のようになって欲しい．また英語の場合には ‘et al.’ でよいが日本語の場合には ‘ほか’ の方がよい．さらに，文献をソートする場合には著者名を主キーとするスタイルが多いので，ここが漢字コードであると漢字コードの順に並ぶので具合がよくない．

alpha では著者の名前からラベルを作るが，英語と混在することを考えるとラベルも

[松篠 87] 松井, 篠原 …

のような形よりは

[MS87] 松井, 篠原 …

のほうが望ましいと考えた．

4.2 日本語用のスタイルの試作

上記のような問題を解決することを考慮したスタイルを標準スタイル plain, alpha, abbrev, unsrt を基に jplain, jalpha, jabbrev, junsrt という形で作ってみた．また情報処理学会論文誌 tipsj, 情報処理学会欧文論文誌 jipsj, 電子情報通信学会論文誌 tieic, 日本オペレーションズリサーチ学会論文誌 jorsj, 人工知能学会誌 jsai, ソフトウェア科学会誌 jssst 用のスタイルも試作してみた．

参考文献のソートあるいはラベル作成を巧く行うことを考えさせて，日本語の文献には著者名のローマ字標記を ‘名前’, ‘姓’ の順で (英語風に) 入れる ‘yomi’ なる新たなフィールドを bib ファイル中に書くこととした^{17,18}．こ

¹⁷ この順番に異論のある人もあろうが [4]，多くの日本人がこの形で (英語で) 書いて論文を発表していることからこう決めた．

¹⁸ ここにひらがなで ‘読み’ をかくことも，もちろんできるから，五十音順にソートすることも可能である．

のフィールドを使ってソートを行ったり, ラベルを作るような形で上記のスタイルを作成した¹⁹,²⁰.

¹⁹ yomi というフィールドを加えるために **bibtex** 本体を変更する必要はない. スタイルファイル中の **ENTRY** に追加するだけである.

²⁰ 著者名の処理以外にも変更を加えている. 詳しくは配布キット中の **jbtxbst.doc** を参照されたい.

jalpha スタイルの一部を例として示す (ただし配布済みのスタイルとは異なる部分がある).

```

FUNCTION {format.names}
{ 's :=
  #1 'nameptr :=
  s num.names$ 'numnames :=
  numnames 'namesleft :=
  { namesleft #0 > }
  { s nameptr "{ff}{ll}" format.name$ is.kanji.str$
    {s nameptr "{ff}{ll}" format.name$ 't :=}
    {s nameptr "{ff~}{vv~}{ll}{, jj}" format.name$ 't :=}
  if$
  nameptr #1 >
  { namesleft #1 >
    {" , " * t * }
    { t "others" =
      { s is.kanji.str$
        {"ほか" * }
        {" , et~al." * }
      if$
    }
    { s is.kanji.str$
      {" , " * t * }
      { numnames #2 =
        {" and " * t * }
        {" , and " * t * }
      if$
    }
    if$
  }
  if$
}
if$
}
if$
}
't
if$
nameptr #1 + 'nameptr :=
namesleft #1 - 'namesleft :=
}
while$
}

```

4.3 データベースの記述方法

以上のようにしたjBIBTEXのためのbibファイルの中身は次のようにする²¹.

²¹ 著者の姓と名の間にスペース (半角でも全角でも) を入れておかないと, jabbrv スタイルの時に姓だけでなく, 名も出力されてしまうので注意が必要である. スペースで姓と名を

```

@ARTICLE{la,
  author="Masao Iri and Kazuo Murota and Shouichi Matsui",
  title="Linear-time approximation algorithms for finding the
  minimum-weight perfect matching on a plane",
  journal=ipl,volume=12,pages="206--209",year=1981
}

```

区切った場合には、スタイルファイルの記述において、漢字コード著者名を `format.name$` して得られる `first name` には姓が、`last name` には名が得られることに注意が必要である (`format.name$` を完全には日本語対応にしていなかったためである。しかし `family name`, `last token` と読み替えれば問題ないし:-)。日本語対応のスタイルファイルを作成した経験から言うと、漢字コード著者名とそれ以外の場合に同じフォーマットを指定して `format.name$` を呼び出すことはほとんどないので、混乱するかもしれないが、あえてこのままにしてある)。また、例では漢字コード著者名の区切りとして、“and” を使っているが、ここはもちろん、“,” (但し全角!) でよい。

```

@ARTICLE{あふれ無し,
  author="松井 正一 and 伊理 正夫",
  title="あふれのない浮動小数点表示方式",
  journal="情報処理学会論文誌", volume=21, number=4, pages="306--313",
  yomi="Shouichi Matsui and Masao Iri",
  year=1980
}

```

このようにデータベースを作成しておくとしplainならば

[1] Masao Iri, Kazuo Murota, and Shouichi Matsui. Linear-time approximation ...

[2] 松井正一, 伊理正夫. あふれのない浮動小数点表示方式. ...

のような形で参考文献リストが得られる. jalphaなら

[IMM81] Masao Iri, Kazuo Murota, and Shouichi Matsui. Linear-time approximation ...

[MI80] 松井正一, 伊理正夫. あふれのない浮動小数点表示方式. ...

となる. jabbrvなら

[1] M. Iri, K. Murota, and S. Matsui. Linear-time approximation ...

[2] 松井, 伊理. あふれのない浮動小数点表示方式. ...

となる. 著者が漢字コードなのに 'yomi' がない場合には, 例えば,

[松井 80] 松井正一, 伊理正夫. あふれのない浮動小数点表示方式. ...

となるようにしてある (著者が複数でも先頭の著者のみからラベルを作るようにしてある). ただし, ソートキーには漢字コード (EUC コード) が使われるから, 順番はおかしなものとなる.

5 おわりに

本稿では日本語化した BibTeX である jBibTeX について述べた. バージョン 0.20 まではとりあえず動くという形で作成してあったが, バージョン 0.30 では十分実用に耐えるレベルに達したと考えている. 日本語用のスタイルについても同様である. 現在までのところ利用者がさほど多くないため, 使用経験が乏しい. 今後は使用経験に基づき, 日本語のために必要な組み込み関数の変更・追加などの改良を加えていきたい. また各種の学会用のスタイルの作成も必要であると考えている. また Oren Patashnik 氏からの私信によれば, 今年中 (1991) に version 1.0 をリリースする予定らしいので, その日本語化も必要であろう.

参考文献

- [1] Leslie Lamport. *LaTeX: A Document Preparation System*. Addison-Wesley, 1986. (邦訳:『文書処理システム (LaTeX)』 Edgar Cooke, 倉沢良一 監訳, 大野 俊治, 小暮 博道, 藤浦 はる美 訳, アスキー, 1990 年).
- [2] Oren Patashnik. BibTeXing. Documentation for general BibTeX users, January 1988.
- [3] Oren Patashnik. Designing BibTeX styles. The part of BibTeX's documentation that's not meant for general users, January 1988.
- [4] マーク・ピーターセン. 日本人の英語. 岩波新書 (新赤版) 18. 岩波書店, 1988.