Njove : JOVEを基にしたUNIX用日本語エディタ*

4F−2

川口 湊†, 織田 誠浩‡, 篠 競§

福井大学¶

1 Njoveの基本的な性格

Emacs系エディタとしての汎用性と高度の操作性とに併せて、特に専門的な使用目的に適した編集機能を兼備した日本語エディタとして'Njove'を作成した。その基盤となっているのは、C言語のみで比較的コンパクトに作られているJOVE[1]である。したがって、JOVEの持つ敏捷性をNjoveもまた継承している。

Njoveの開発は、電子総合技術研究所におけるNemacs[2]の開発とほぼ時を同じくしている[3][4]。基盤となっている'Emacs'の特性を活かすという基本的な設計方針も両者に共通しているので、ごく自然の帰結としてNjoveとNemacsの間には類似点が多い。しかしながらNjoveは開発の当初から、汎用のツールとしてのエディタに要求される使いやすさを追求する一方で、研究活動を支援するシステムの重要な一翼を担うものと位置づけており、単にJOVEを日本語化することに止まらず、大規模ソフトウェアの開発や、専門的な著書や論文等の執筆などの、高度の編集作業の遂行に有効と考えられる機能を新たに組み込むなどの拡張を行ってきた。

以下にNjove独自の機能を中心にそのうちのいくつかを紹介する。他にも例えば執筆支援機能として、JOVEにはないoutlineモードをはじめ、第二水準までの全ての漢字を柔軟に検索・表示し入力する機能、英和辞典や日本語の用語辞典を参照する機能、使用者が指定する多数の'マニュアル'ファイルを参照する機能、なども追加した。

また、Njoveそれ自体とは独立のものであるが、かな漢字変換のためのwnn (uum) とNjoveとの相性を改善してtouch typingを可能にするためのプログラムとして'pwnn'を作成し、Njoveと組み合わせて常用している。

2 欧文および日本文の一体化

専門的な内容の文書ファイルでは、日本文と欧文と混在している場合が少なくないことから、欧文と日本文とを完全に一体化して編集を行うことができる必要がある。当然ながら'バイト'ではなく'文字'を最小操作単位とし、'Emacs'で頻繁に用いられる単語 (word) の概念を最も合理的な方法で日本文にも適用する。日本文字については、かたかな、ひらがな、漢字、およびそれ以外の全角記号、のそれぞれを異なる字種として区別し、同一字種の並びを一つの (編集操作の対象としての) '単語' として扱う。

3 自動右揃え

Emacs系のエディタではauto-fill modeを選定してfill-paragraphを実行すると、画面の右端で最初にはみだす単語の直前の境界 (word delimiter) の位置で自動的に折り返す。一般に任意の文字の境界で折り返す習慣のある日本語の文書の編集では、画面の表示の右端を揃えることで表示画面の可読性がかなりの程度向上する。したがって、エディタにおける編集作業の生産性を高めるうえで画面の右端を折り返すことが欧文の場合にもまして重要である。

そこでNjoveでは副モードとして、auto-fill modeとは独立に"wrap-around mode"を新たに設けた。wrap-around modeでは、1文字を挿入または消去する毎に該当する段落全体にわたって右端の自動折り返しを実行する。英文の部分については単語の境界で折り返し (word wrap) を行う。日本文字の部分は実質的な右揃え (right justification) をして表示する。

見かけ上の単純さとは異なり、wrap-around modeで行う折り返しの処理は相当に複雑なものとなる。例えば、auto-indent mode自体がかなりデリケートな処理を行っているが、それとwrap-around modeとが併用される可能性があることから、段落全体の状況を常に正確に把握しながら字下げの部分の空白の処理を実行し、行頭・行末禁則にも正しく対処して折り返しを行うようにしている。

ところで、Emacs系エディタの長所の一つが表示画面とファイル (バッファ) の内容とが厳密に対応していることである。そこで、fill-paragraphコマンドにおいてもまたwrap-around modeにおいても、右端での折り返しのために必要であればNjoveはバッファに対して随所でnewlineを挿入または削除する。ところが、編集対象のファイルによっては特定の位置のnewlineコードの有無について恣意性が許されない (例えばTEXファイルにおけるコメント行) 場合がある。そこで基本的に相反するこれらの条件を両立させて利用者の意図を常に正しく反映できるようにする設計が要求される。

---

*Njove : a Japanese-Language Editor for UNIX based on JOVE

†Minato KAWAGUTI
‡Masahiro ODA
§Kisou SHINO
¶Fukui University

4 多数のファイルの一括編集

比較的規模の大きなソフトウェアや、TEXで書かれた著書の原稿などは多数のファイルの集合より成り立っているのが普通である。これらの多数のファイルの集合の全体を一括して文字列の探索および置換といった編集作業を行うのに好都合なように、副モードとして“global edit mode”を導入した。多数のファイルを編集の対象としてそれぞれのバッファに割り当てた後、それぞれのバッファ毎にそれをglobal editの対象とするか否かを随時選択できる。global-edit modeを選択すると、global editを許すバッファのみを並べて一続きのものとしたものの全体を対象に、(1)探索コマンド(通常の探索とインクリメント型探索のそれぞれについて前後方の4種類)、(2)置換コマンド(単純置換および照会型置換の2種類)、および(3)mark位置へのポイントの移動、を実行する。

global-edit modeは特にバッファの数が多くなって全体の把握が困難になる事態が想定されるので、使用できるコマンドの種類とその適用範囲に制限を設けている。これにより、コマンドの誤入力などによる不測の事態でのバッファの内容の破壊などの危険が不用意に広域に及ぶことを未然に防ぐ。例えば、pointおよびmarkに関連したコマンドはglobal-edit modeでは“global point”および“global mark”を対象とするように切り替わり、それまでのバッファ毎の通常のpointおよびmarkの位置はglobal-edit modeが選択されている間は凍結されたままになる。markとglobal markとは別個のring bufferに記録されるから、互いに独立していて干渉しあうことはない。

また、領域(region)や文(sentence)は個々のバッファの内部に限定している。したがって、領域あるいは文を有効域とするコマンド、例えば消去コマンドがバッファの境界を超えて二つ以上のバッファにまたがって作用することはない。

5 TEXファイルの編集

特に学術文書ではTEXが日常的に使用されているが、TEXが使用するコマンド体系は膨大かつ繁雑であるので、TEXのファイル作成の効率を挙げるにはエディタに適切なTEXファイル編集支援機能を盛り込み、それを活用することが重要な鍵となる。そこで次のような機能を付加した。

5.1 TEX mode

一部のNjoveの機能はTEXファイルの編集の際には言語依存型の特有の振る舞いをするように設計しておくと便利であるのでJOVEには欠落していたTEX modeを新たに設け、`\latex-close-block`のようなEmacsに備えられているTEX modeで使用できる便利なコマンドをNjoveにも付加した。

5.2 TEXのコマンド名の確認

plain TEXは約900種類のTEXコマンド(TEX macro)とTEX変数が定義されている。これに加えてTEXコマンドとは微妙に異なる多数のLATEXコマンドとLATEX変数がある。特にplain TEXの利用者にとってはこれらのコマンド名および変数名を正確に記憶することは容易ではない。そこで、apropos機能に倣った‘tex-apropos’によりTEXおよびLATEXのコマンドおよび変数の綴りを確認したり関連したコマンドや変数を簡便に探索したりできるようにした。

5.3 TEX modeでのword abbreviation

TEXファイルを作成する速度を向上し併せてコマンドの綴りの誤りなどを避けるために特に有効な手法の一つは、頻繁に使用するTEXコマンドに対してTEX modeにおいてのみ有効な略称コードをword abbreviation modeで展開させる機能を活用することである。これには、入力ミスを検知するのを容易にするという別の利点もある。Njoveのword abbreviation機能は略称コードを単に登録された文字列に展開するだけに止まらず、TEX modeでは、もし展開すべき文字列の中に中身のない中括弧の組“{}”が含まれていれば、左から順に中括弧の内部に自動的にポイントを移動してそれぞれの中括弧の部分に文字列を挿入することを促す。

5.4 TEXマクロの定義部分の参照

UNIXのツールであるctagsに倣って“`.tex`”を拡張子とするファイルに対してのみ選択的に機能する‘dtags’を作成した。これを用いてctagsの場合と同様に予めtagsファイルを作成しておくと、(1) `\def`、`\gdef`、または`\xdef`で定義されているマクロ、または(2) `\font`で定義されているフォント、の定義部分を含むファイルの該当行に簡単にポイントを移動することができる。

参考文献

[1] Jonathan Payne: *JOVE Manual for UNIX Users*, 4.3BSD版マニュアル (1986)

[2] 半田 剣一, 小方 一郎: *Nemacs*, **bit 20** 1114–1123 (1988)

[3] 川口 湊, 鈴木 重治, 藤井 弘, 吉田 肇: *Njove : 日本語機能を付加した UNIX実時間エディタ JOVE*, 福井大学工学部研究報告 **36** 263–272 (1988)

[4] 川口 湊, 鈴木 重治, 藤井 弘, 吉田 肇: *JOVEの日本語化*, **jus** 12th UNIX Symposium Proc., 90–99 (1988)