

OS/omicron 上の日本語エディタ 4N-5

小林伸行, 並木美太郎, 高橋延匡
(東京農工大学)

1. はじめに

我々の研究室では、レーザービームプリンタを用いた日本語フォーマッタ (JOSH0), オンライン手書き文字認識 (JOLIS) などの日本語情報処理を前提としたオペレーティングシステム OS/omicron の研究、開発を行っている。

本稿では、OS/omicron 上で動作する日本語エディタの開発と、日本語プログラミング環境としてのエディタの利用について述べる。

2. エディタの開発方針

- 本エディタは、次のような開発方針をとった。
- (1) 利用者としてある程度エディタに慣れている者、おもに当研究室の学生を想定した。
 - (2) 複数のハードウェア上で動作するように、OS やハードウェアにできるだけ依存しないように考慮した。
 - (3) 処理対象として日本語を含むプログラムから文書までを想定し、日本語の容易な入力、編集を可能にした。
 - (4) キーへのコマンドの割当てやコマンドの追加をユーザに解放し、自由な定義を可能にした。
 - (5) OS のマルチタスク機能を利用し、コマンドインターフリタや言語処理系などを呼び出すことにより、日本語プログラミング環境を提供した。

3. 日本語変換機能

OS/omicron は、より自然な形で日本語処理を行うために、全 2 バイトの文字コード体系 (JIS X 0208 準拠) を用いている。また、OS/omicron 上でソフトウェア開発に用いられる言語 C コンパイラ (CAT) は、識別子にも日本語を許している。このため、日本語がプログラム中にも頻繁に用いられ、日本語を容易に入力できることが重要となる。

本エディタでは、OS/omicron の仮名漢字変換システムを利用し、モードレスな日本語変換を実現した。これにより、自然な日本語の入力が可能である。

4. エディタへのコマンドの入力

エディタには各種のコマンドがある。これらのコマンドをどのように呼び出すか、また必要なコマンドが備わっているかにより、その使い勝手は大きく左右される。

本エディタは、キーへのコマンドの割当てをユ

ザに解放している。モードやプレフィックスは割当て表を切り替えるコマンドとして定義され、エディタのコマンドは日本語変換も含めて、それ自身にはモードを持たない。このため、ユーザはモードやプレフィックスを自由に組み合わせて、自分の好きな配列に変えることができる。

また、既にあるコマンドを用いて新しいコマンドを作成することができる。ユーザは、言語 C でコマンドを組み合わせて新しいコマンドを作成、コンパイルし、エディタのライブラリとリンクする事で、コマンドを追加できる。言語 C の構文が利用できるので、エディタの内部状態に応じた細かい定義が可能である。

上記の機能により、ユーザは固有の操作系を構成できる。

5. マルチウィンドウ、マルチバッファ

本エディタは、マルチウィンドウ、マルチバッファを実現している。このため、複数のファイルを同時に編集したり、1つのファイルの内容を2つ以上のウィンドウで編集することが可能である。マルチウィンドウはオーバーラッピング方式であり、汎用のキャラクタディスプレイでも表示可能である。図 1 に、本エディタの実際の動作画面の例を示す。

また、OS のマルチタスク機能と組み合わせることにより、エディタを他のアプリケーションの入出力インターフェースとして使用することができる。たとえば、コマンドインターフリタをエディタのウィンドウに割り当てることで、言語 C のコンパイラや日本語を扱うアプリケーションなどを呼び出し、その入出力の結果をエディタのウィンドウに表示させることができる。

この機能により、ファイルの編集が容易になるだけでなく、エディタをプログラミング環境の中核として使用できる。

6. アンドウ

ファイルの編集中にアンドウが必要となる場面は比較的多い。たとえば大域的に誤った置換を行なった場合などを考えると、必要不可欠な機能であるといえる。

本エディタでは、ほとんどのコマンドに対してアンドウが可能である。アンドウは複数回実行でき、有効回数はユーザが指定できる。また、実行したアンドウを逆順に戻すリドウも備わっている。

The screenshot shows a terminal window for the OS/omicron editor. The title bar reads "OS/omicron" and "第18回 JOSH0・JOLISミーティング議事録". The date "87-11-10 本間▼" is displayed. The window contains a list of meeting minutes and a transcript of a discussion. The transcript includes messages from participants like 鈴木, 森, 中川, and 高橋. The source code visible at the bottom is in C, dealing with file operations and buffer management. The right margin of the window has several status indicators and command keys.

```

OS/omicron
第18回 JOSH0・JOLISミーティング議事録      87-11-10 本間▼ -----*/▼
1. 日 時 1987年 10月 27日 15:20 - 16:00▼          */▼
2. 場 所 5 - C▼          */▼
3. 出席者 高橋 中川 #002議事録          */▼
    曽谷 皆川 そうと思う。▼
    横関 (敬) 鈴木: コンパイルしてやってもうまくいくのではないか。▼
4. 議 題▼ 森: プリミティブがどこまで書いていいかという情報をしきると思う。▼
    (1)図形作成言語の設 中川: プログラムが1つ1つ完結していれば、それでいいかもしれない
5. 資 料▼ インタプリットしてどうしても遅くなりそうだというのなら、考
    (1)図形作成言語の設 中川: 言語Cは拡張性がある。図形専用のフレームメモリを扱う関数
        自分で書いたプログラムとリンクすれば、コンパイラができる。
6. 内 容▼ 高橋: それを使うためにはオブジェクトコードだけ出して、別のフ
    まえばよいのだ。▼
6. 1 前回会議事録 鈴木: パーザとコードジェネレータを使わなければいけないが、そ
    鈴木未来子より前回 チタスクでなければいけないのか。▼
6. 2 図形作成言語の 設議事録 高橋: シングルタスクをシーケンシャルにやればよい。▼
    FDCL_GLBL INT 田中: CP/Mだとオーバレイができない。▼
    FDCL_GLBL AVOID 中川: OS/oができる前に、トランスレータとマクロプロセッサをき
    FDCL_GLBL INT ように。▼
    FDCL_GLBL AVOID #002議事録
    FDCL_GLBL AVOID thining(); /* 間引き処理 */

/* 外部参照変数の宣言 */
DDCL struct k_sou_data sou_data[]; /* 原データバッフ
前処理.c

```

図1 日本語エディタの動作画面の例

7. エディタの大きさ

本エディタは、OS/omicron上だけでなくMS-DOS上でも動作している。現在、コマンド数は約120個(表1 参照)，ソースプログラムは言語Cで約1万行、実行モジュールはOS/omicron上で約70KByte、MS-DOS上で約40KByteとなっている。機能の追加は現在も行なわれている。

8. おわりに

本エディタを実現したことにより、OS/omicron上で日本語を含むプログラムや文書を入力、編集するための手段が得られた。さらに、コマンドインタプリタや言語処理系、日本語を扱うアプリケーションなどを統合し、日本語情報処理を行なう環境を提供した。

参考文献

- [1] 小林伸行, 中川正樹, 高橋延匡: “OS/o上の日本語エディタ”, 情処学第36回全大, pp.343-344 (昭 63)
- [2] 鈴木茂夫, 小林伸行, 田中泰夫, 中川正樹, 高橋延匡: “OS/omicronにおける日本語プログラミング環境”, 情処学コンピュータシステムシンポジウム, pp.11-18 (昭 62)

表1 エディタのコマンド(一部)

コマンド名	処理内容
Abort_Command	実行中のコマンドを中止する。
Auto_Indent	自動的にインデントーションを行なう。
Beginning_Of_Buffer	バッファの先頭に移動する。
Beginning_Of_Line	行の先頭に移動する。
Beginning_Of_Paragraph	段落の先頭に移動する。
Change_Map	キーへのコマンドの割当て表を変更する。
Check_And_Exit	ユーザに確認し、エディタを終了する。
Check_User	ユーザに確認する。
Clear_Message	メッセージをクリアする。
Clear_Undo_Stack	アンドゥスタックをクリアする。
Close_Bind_Mark	指定ウィンドウに付けたマークを閉じる。
Close_Bind_Window	指定バッファの表示ウィンドウを閉じる。
Close_Buffer	指定バッファを閉じる。
Close_Mark	指定マークを閉じる。
Close_Window	指定ウィンドウを閉じる。
Connect_Line	行を連結する。
Delete_Next_Character	カーソル直後の文字を削除する。
Delete_Paragraph	カーソルのある段落を削除する。
Delete_Previous_Character	カーソル直前の文字を削除する。
Delete_Region	指定マークとカーソルの間を削除する。
Delete_String	指定した長さの文字列を削除する。
Divide_Line	行をカーソル位置で分割する。
Down_Window	指定ウィンドウを下に動かす。
Draw_Display	画面表示を行なう。
End_Of_Buffer	バッファの末尾に移動する。
End_Of_Input	入力文字列を決定する(文字列入力時)。
End_Of_Line	行の末尾に移動する。
End_Of_Paragraph	段落の末尾に移動する。
Exchange_Point	指定マークとカーソルの位置を交換する。