

3N-2

プログラマのレベルに対応した
ワークステーションエディタ

竹田 学

(富士通 ㈱)

1. はじめに

汎用フルスクリーンエディタ（以降、単にエディタと呼ぶ）は、プログラム開発をする上でプログラマが最も多くかつ長時間使用するツールの一つであり、プログラム開発の作業効率に影響する重要な役割を持っている。

これまでワークステーションのエディタは、初心者向き、熟練者向きというように特定のレベルのプログラマを対象としたものが提供されていた。

しかし、エディタを使用するプログラマは、計算機に対する知識の違い、技術力の違い、経験の違い、打鍵速度の違い等によりさまざまなレベルで存在しており、これらすべてのプログラマを満足させるためには、各レベルのプログラマに合った機能および操作方法を持ったエディタが必要である。

ここでは、これらさまざまなレベルのプログラマが満足でき、かつソフトウェアの開発効率を向上させるエディタをワークステーション上で実現するには各レベルのプログラマに合った機能および操作方法をどのように統合すればよいかについて説明する。

2. プログラマのレベルとエディタに必要な機能

プログラマのレベルを初心者、中級者、熟練者に分け、各々のレベルのプログラマに対してエディタに必要な機能および操作方法を次に挙げる。

a) 初心者に対して

- 簡単な編集操作。
- マニュアルなしでも操作できる。
- システム内で一貫した操作方法。

b) 中級者に対して

- キーボード主体の編集操作。

c) 熟練者に対して

- 高速編集操作。
- 豊富なカスタマイズ機能。
- 高度な編集機能。

f) 各レベル共通

- 日本語入力機能。
- リカバリ機能。
- 文字の多国語化

3. 実現方法

前節で述べた各プログラマのレベルに必要な機能および操作方法を実現するには具体的にどのようにすればよいか述べる。

a) 簡単な編集操作

マウス、メニューだけで編集操作ができるようにする。

一般的にワークステーションでの操作は、マウス、メニューによるものが多い。また、このほうがコマンド名や対応するキーを覚える必要がなく初心者にとっても修得しやすい操作である。

b) マニュアルなしによる操作

エディタの機能および操作方法の説明をメニューで表示させる。

c) システム内で一貫した操作方法

例えば、次のものがある。

- コマンド名の統一。
- マウス入力に対する機能および操作方法の統一。
- ファンクションキーの意味および機能の統一。
- メニュー形式の統一。

d) キーボード主体の編集操作

より速い操作ができるようファンクションキーにコマンドを割り当て、マウスなしでも編集できるようにする。

さらに、ラインコマンド入力機能を付加する。

ラインコマンドとは、削除なら“DELETE”というようにコマンドを文字列化させ、さらに処理に必要な情報をオペランドとして同時に入力できる編集操作のことである。

Workstation Editor for various programmer's level

Manabu Takeda

FUJITSU, Ltd.

この機能に付随してテキストの行番号の表示機能も必要である。

e) 高速編集操作

二重打鍵編集操作を付加する。二重打鍵編集操作とは、編集画面上でCONTROLキーと英数字キーを同時に押下することにより、処理範囲選択とコマンドを一度に実行する操作方法のことであり、ホームポジションでの操作ができる。

さらに、必要な項目として、高速画面スクロール／カーソル移動の機能が必要である。

f) 豊富なカスタマイズ機能

- － ファンクションキーのコマンド割り当てを変更できること。
- － 二重打鍵コマンド名を変更できること。
- － 編集環境設定項目を充実させる。

g) 高度な編集機能

例えば次のような機能がある。

- － 画面分割あるいはマルチウィンドウによる複数ファイル同時編集機能。
- － 矩形によるカットアンドペースト機能。
- － 16進表示による編集機能。
- － 特定行／けたの非表示機能。
- － 階層化したUNDO／REDO機能。
- － 探索文字列の正規表現機能。

h) 日本語入力機能の付加

編集画面に直接日本語を入力する機能、単語登録／抹消機能が必要である。

i) リカバリ機能の付加

編集中のシステム異常あるいは電源断等によりエディタが終了した場合でも編集中のテキストを可能な限り復元させる。

j) 多国語化対応

表示されるリテラル項目を各国の言語で外付け化し、静的または動的に切り替えできるようにする。

その他、プログラマには、今まで使用していたエディタの操作方法が身についている者がいる。これらのプログラマのため、そのプログラマが使用していたエディタに対して違和感のない操作方法を付加することも必要である。

さらに、熟練者レベルのプログラマでは、エディタ主導型のプログラム開発体系が必要である。すなわち、編

集中のソーステキストで構文チェック、デバッグ、翻訳する機能というようなプロダクト間連携機能である。

エディタの実現方法の概念を図1に示す。

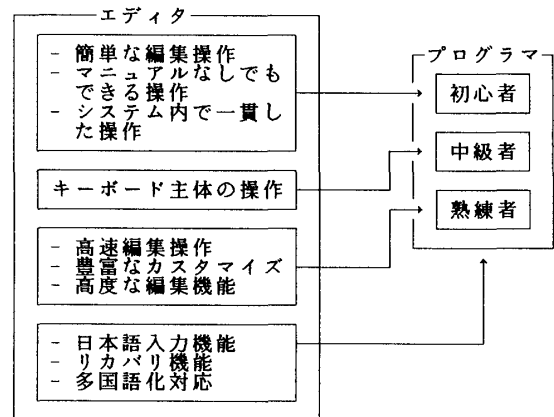


図1 エディタの実現方法の概念

4. おわりに

ここで説明したエディタをさらに発展させ、次のような機能を持ったエディタの作成を今後の予定としている。

- a) プログラマの打鍵速度、操作のミスの回数、コマンド使用率等を記憶し、徐々に高度な機能を付加していく学習機能。
- b) プログラマによっては偏った使い方をする場合がある。そこで、エディタの機能を細かく部品化し、プログラマ自身が自分に合ったエディタを作り上げることができる部品化機能。