

# OS/omicon 日本語コマンドインタプリタにおける 4N-4 コマンド入力系

本宮志江, 並木美太郎, 高橋延匡  
(東京農工大学 工学部 数理情報工学科)

## 1. はじめに

当研究室では、日本語情報処理のためのオペレーティングシステムOS/omiconを研究、開発中であり、現在第2版が動作している。OS/omiconでは、日本語の使用を前提とした全2バイトコードを採用している。またシステム記述言語である言語Cコンパイラcatも全2バイトコードとなっている。このため、自然にコマンド名、ファイル名などに日本語を用いることができる。このような日本語環境のもとに、日本語入力およびそれに関する変換機能(仮名漢字変換など)を持つ日本語コマンドインタプリタが実現されているが、実際には一部のファイル名とソースプログラムにのみ日本語が使われているという状況である。コマンド名、ファイル名を日本語にすることによって、コマンド名からコマンドの機能を、ファイル名からファイルの内容を連想することが容易になるということが期待できる反面、打鍵数が増加し、慣れたユーザには使いにくくなる。そこで本稿では、コマンドインタプリタのコマンド入力系を検討し、効率的な入力の一方式を提案し、その設計方針を述べる。

## 2. 基本方針

日本語コマンドインタプリタという限り、日本語のコマンド名、ファイル名を標準として扱いたい。しかし、日本語のコマンドには様々な問題があり、日本語入力機能を持ちながら、コマンドは依然として英語となっている。そこで、現在あるコマンドインタプリタに変更を加えることにより、一番の問題となるであろう入力系について検討し、漢字、仮名を使ったコマンドでありながら英字のコマンドと同等の手間で入力できる方法の一つ確立することを本研究の第一の目的とする。

そして、最終的な目標は、“OS/omiconにおける日本語プログラミング環境の研究および評価”である。

## 3. コマンド入力系の設計

我々の設計するコマンドインタプリタと入力系は、OS/omiconユーザとしてプログラムを作成した経験のあるものとして設計を行う。

### 3.1 コマンドの日本語化とその問題点

現在のOS/omiconユーザは、現存のコマンドインタプリタのコマンドに慣れてしまっている。そのために、

コマンドが日本語になることは単に煩わしいことになりかねない。その理由としては、

- (1) 今までのコマンド名と日本語のコマンド名の対応を覚えなくてはならない。
- (2) 日本語(漢字、仮名)を使用するという事は、入力作業に変換という手間が加わる。などが挙げられる。(1)に関しては、名前のつけ方次第で、逆に利点(わかり易さ)にもなると考えられるし、それで不十分な点はコマンドの仕様などを表示するヘルプ機能でカバーできると考えている。また、現実の日本語ワードプロセッサでは、日本語のコマンドを使用している現状もある。したがって、いかに日本語のコマンド入力を簡単にするかが第一の問題となる。

### 3.2 コマンド入力系

ここでいうコマンド入力系とは、以下に示すようなものである。

- (1) コマンドの入力方式(デバイス)
- (2) コマンドの入力形式
- (3) コマンド名
- (4) 引数の置き方
- (5) コマンド名の登録と検索方法

入力方式にはキーボード入力を採用する。この方式は現在主流であるし、現在当研究室にあるコマンドインタプリタもキーボードによる入力である。今回は実験の意味も含んでいるし、すぐに使用可能なキーボードのみを用いる。結果によっては、タブレットの使用も考えている。

入力形式は、コマンド名の読みを全部入力することを基本とする。なお、コマンド名の読みはローマ字表記でなくてもよい。コマンド名には標準形を設定し、コマンドインタプリタの方ではその名前のみを持つが、入力は、コマンド名の読み全部の他に、キーワードを設定することにより、コマンド名の読みの一部や英字一文字(コマンド名の頭文字など)でもコマンドの指定ができるようにする。コマンドインタプリタの内部コマンド、外部コマンドに関しては、予め表1に示すように3通りの名前を用意しておくが、ユーザの好みで各ユーザがそれぞれに名前をつけ、その名前でもコマンドを指定することもで

きるようにする。また、新しく作ったコマンドには、日本語と英語の両方の名前をつけてもらう。

コマンド名は、何よりもわかり易さを重視する。ユーザ側からすると、そのコマンドが何をやるものであるのかが、わかるものでなくてはならない。逆に、作成者側からすれば、不正確な英語でコマンド名を考えるよりも、使いなれた日本語で名前をつける方がはるかに簡単である。先に述べたように、コマンド名の読み全部を入力しなくてもコマンドの指定ができるのであれば、コマンド名に下手な省略形を用いなくても入力の手間は少なくできるのである。

入力系の一要素として示した引数の置き方については、日本語の特徴を生かした方法（例えば postfix）があると思うが、今回は英字のコマンドに準ずる（prefix）こととする。

以上述べてきたコマンドは、全て変換表に登録しておく。変換表に登録するのは、キーワードとそれに対応するコマンド名の標準形である。変換表はコマンドインタプリタ内部に持つのではなく、ユーザ拡張部にコマンドインタプリタ用のユーザ辞書という形で持ち、それを仮名漢字変換の変換機能を通して取り出す。これと同様にして、各ユーザはユーザ用にコマンド辞書なるものを作り、そこに新しく作ったユーザ専用のコマンドやユーザ用に名前をつけ直したコマンドなどを登録しておくことができる。変換の際の辞書の検索順は、

ユーザ専用辞書

↓

コマンドインタプリタ用辞書

↓

日本語システム辞書

とする。

表1に、コマンド名の対応表を示す。

まず、表1において、'日本語コマンド名'は、コマンドの機能を日本語でわかり易く示した名前であり、コマンドインタプリタが内部に持つコマンド名の標準形である。'キーワード'は、入力の手間を省くためのコマンドの別名である。つまり、コマンド名の読み全部の入力、キーワードの入力のどちらでもコマンドの指定ができる。また、'英字コマンド名'は、現存のコマンドインタプリタのコマンド名である。

つぎに、1つのキーワードに対して複数の候補がある場合の選択方法を説明する。例えば、ディレクトリに関する操作には、rmdir, mkdir, dir, cdがある。'ディレクトリの作成'をしたいときには、以下のような操作をする。

```
d [変換] キー
  ↓
ディレクトリの削除 [変換] キー
  ↓
ディレクトリの作成 [CR] キー
```

ユーザは目的のコマンド名が表示された時点で確定する。このようにしてコマンド名の指定を行うが、複数候補の表示の順序は現在のところ変更できないようになっている。

表1. コマンド名対応表

#### <内部コマンド>

日本語コマンド名 (標準形)	キーワード	英字コマンド名
コマンドインタプリタの終了	c command	logout
ディスクの残り容量の表示	d disk	status
ディスマウント	d dis	dismnt
ディレクトリの削除	d directory	rmdir
ディレクトリの作成	d directory	mkdir
ディレクトリの表示	d directory	dir
ディレクトリの変更	d directory	cd
ファイルのコピー	f file	copy
ファイルの削除	f file	era
ファイルの内容の表示	f file	type
ファイル名の変更	f file	ren
フラッシュ	f flush	flush
時間の設定と表示	j jikan	time
マウント	m mount	mnt
パスの設定	p path	path

#### <外部コマンド>

日本語コマンド名 (標準形)	キーワード	英字コマンド名
アセンブラ	a assembler	asm68000
コードジェネレータ	c code	fcodegen
コマンドインタプリタ	c command	cli.trap8
コンパイル	c compile	ccat.sub
コンフィグ	c config	config
ディレクトリコピー	d directory	directory
ダンプ	d dump	dump
エディタ	e editor	alt
ファイルコンバータ	f file	msdos
フォーマット	f format	edofrmt
フロッピーディスクの初期化	f floppy	fsysinit
リンク	l link	link.sub
文字列の検索と表示	m mojiretu	find
パーサ	p parser	ncparser
プリプロセッサ	p prepro	cpp

#### 4. おわりに

本稿では、日本語のコマンドを効率的に入力できるコマンド入力系を提案した。今後は、入力の問題がなくなったときに、“日本語のコマンド名はユーザにどのような影響を与えるか”を調べるのが課題である。

#### 5. 参考文献

[1] 鈴木茂夫, 皆川良明, 中川正樹, 高橋延匡: “OS/ο用コマンドインタプリタ”, 情報処理学会第36回全国大会 (1988)