香川大学経済論叢
第63巻 第2号 1990年9月 31-64

言語学研究へのパーソナルコンピュータの応用

本田道夫*
山田勇**

I はじめに

言語学研究における計算機の利用はかなり以前から行われ，単語の使用法などを調べた文体の研究などがよく知られている。言語学の研究において対象とする作品が大作の場合には，ある単語の出現位置を手作業で探すとすれば大変な作業となる。しかし，大作といえどもファイルに入力されていれば，計算機を用いることにより迅速な調査が可能であり，その効果は顕著であったと思われる。しかし，従来は，計算機の処理能力などから，それなりの計算機環境の整った研究者のみが可能であった。

* 香川大学経済学部，**香川大学教育学部

一方，最近では，パーソナルコンピュータ（以後パソコン）あるいはワードプロセッサ（以後ワープロ）の価格も低下し，個人あるいは個人研究費でも購入可能となってきた。そして，文章・論文の作成にパソコンを利用する言語学の研究者も増加してきている。さらには，これらのパソコンを紙・ノートおよび鉛筆の代わりに用いたり，さらにデータベースの機能や各種のプログラムを作成して言語学研究の補助に用いている研究者もいる。著者の一人である山田もロシア語の研究・教育にパソコンを利用しようと考えており，パソコンでのロシア語の表示・印刷に関する相談を本田が受けたことが，本論文で述べるシステム開発のきっかけである。最初は，ロシア語の表示・印刷が可能なテキスト・エディタ（以後エディタ）および印刷ソフトウエアの開発に始まり，徐々に以下で述べる言語研究のためのソフトを開発してきた。

言語学の研究・教育の補助として利用するシステムといっても，対象とする言語や利用目的により，必要とされる機能は大きく異なる。たとえば，基本的ともいえる紙・ノートおよび鉛筆の代わりになる程度のことでも，ある言語にとって最適な文字コードの体系が，別のアルファベットを有する言語にとっては必ずしも最適な体系とはなりえない場合もあるし，入力方法についても同様な場合がある。したがって，まず言語を特定し，さらに目的を明確に定めたシステム開発が必要であり，われわれはまず最初の目標として以下のことを設定した。

(1) ロシア語の研究・教育の補助として用いることができるシステムとする。

(2) ただし，ロシア語のアルファベット（ロシア文字）だけでなく，ロシア語特有の特殊記号を扱うことができるシステムとする。

(3) なお，研究・教育においては，当然ロシア語以外にも，日本語および英語も扱えることが必要であるので，これら3カ国のアルファベットが混じった状態で利用できるシステムとする。

(4) 紙・ノートおよび鉛筆の代わりとなる機能は，基本的にはエディタ程度のものでもよく，必要があれば開発する。ワープロほどの文字修飾機能は持たせなくてもよい。

(5) ただし，印刷プログラムを工夫することにより，下線や強調程度の文字修飾の印刷，および英語とロシア語はイタリック体での印刷が可能とする。

(6) ロシア語の研究・教育のための単語帳や熟語帳——テキスト中での各単語あるいは熟語のすべての出現位置とその意味をアルファベット順に整列したもの——を作成するためにも用いる。

(7) ブルガリア語の活用辞書を作成するためにも用いる。ただし，現時点では，まず単語から活用例文を検索するためにデータベース機能を用いることを考えている。なお，ブルガリア語はロシア語と同じアルファベットである。

(8) CRT 画面への表示やプリンタへの出力などもロシア文字でおこなわれるようにする。

(言語学研究の文献など([3])によれば，たとえば，通常の英文字アルファベット以外の文字のときには;a のように「;」を特別のものとして用いて，入力・表示するものもあるようである。しかし，入力の簡素化および本来の字形での表示は効率よく作業を進める上で是非とも必要であると考えた。)

そして，これらの目標を満足するシステムを，市販のソフトウエアを組み合わせて実現することが可能かどうかを検討した。しかし，以下の「II 市販ソフトウエアの現状」で述べる理由から，市販のソフトウエアだけで必要とする機能すべてに渡って整合性のあるシステムを構築することは不可能であると判断した。そこで，必要なものから順次プログラムを開発することとし，基本的機能であるエディタとプリンタへの出力の機能から実現し，さらに，(6)〜(8)と徐々に言語学研究のための環境の整備を図り利用してきた。なお，これらの開発にとりかかる前に決定しておくべき重要事項として，

(1) ロシア文字および特殊文字に対する文字コードの決定

(2) ロシア文字および特殊文字に対するキーボード上のキー割り当ての決定

があったが，その詳細については，以下の「III 文字コードの決定とキー割当の決定」で述べる。

なお，われわれのシステムは，まだ完全に完成しているわけではない。しかし，今後のシステム開発は現在の延長線上に位置するであろうし，しかも，基本的機能の設計・実現が終了していることもあり，開発したシステムの紹介の意味で，現時点までのシステム設計の方針・成果をまとめることにした。したがって，本論文では，個々の機能の設計の方針などについては説明するが，実現方法の技術的な解説は行わない。

ここで設計・開発しているものはロシア語，ブルガリア語などのスラブ系言語を扱うことを目的としており，他の言語に対してそのまま直接利用することはできない。しかし，他の言語に対しても，文字コードおよびキーボードとの対応の決定と，その言語の字体の作成を行えば，同様な機能のシステムを実現することは可能である。スラブ系言語の場合，英語の字体と異なるものが多いが，他の言語では異なる字体は少数であり，比較的簡単に実現可能である。そのような事情もあり，以下では，特にロシア語にこだわらない場合には，日本語あるいは英語以外の言語を，第3国語と呼ぶことにし，第3国語の文字を第3国文字ということにする。

II 市販ソフトウエアの現状

2.1 第3国語の文章の入力と表示の現状

最近のソフトウエアには充実したものが多く，ワープロソフトやPDS (Public Domain Software) などの中には，日本語や英語だけではなくギリシャ語，一般的な発音記号，数学で用いる特殊な記号も扱えるようになっており，通常の文章や論文などを作成するには十分であるものが多い（1）。したがって，システム開発の前に，まず，そのようなワープロ機能を基本的な入力機能とし

（1） このようなものとしては，「ジャストシステム」の『一太郎』[9]，「高電社」の『Techno Mate』[10]，「岩波書店」の『SPE』[4]，「イシガキ・エム・イー・エス」の『bits』[8]，PDSの『Myfont』などがある。これらの市販のものについては，経済学部の先生方が購入されたソフトウエアのマニュアルで調べ，若干の不明な点については，確認のために利用させていただいた。なお，調査・検討以降にバージョンアップされ機能強化が図られたものがあるかもしれない。

て利用する可能性を検討した。

市販あるいはPDSのソフトウエアでは第3国語の文字のコード割当，表示，およびキーボードからの入力は大別して次のような方法がとられている。もちろん，これらのコード割当，表示，キー入力は互いに密接に関連しており，以下の任意の組み合わせが可能なわけではない。

【コード割当】

(a-1) MS-DOSおよびパソコンのユーザ文字定義の2バイトコードに割り当てる方法

(a-2) ワープロソフト独自の2バイトコード体系を用いる方法

(a-3) カタカナの使用は半角カタカナを用いずに全角のカタカナを用いることにし，半角カタカナの1バイトコードを第3国語の文字に割り当てる方法

【表示】

(b-1) MS-DOSのユーザ定義文字の機能を用いて全角文字で表示する方法

(b-2) ワープロ独自の外字登録機能を用いて，グラフィック画面に半角文字として表示する方法

(b-3) MS-DOSのCRT表示のソフトウエア割り込みの部分をトラップし，グラフィック画面に半角文字で表示する方法

(b-4) MS-DOSのユーザ定義文字の機能と，(1-2)と同様に画面表示のソフトウエア割り込みのトラップを組み合わせて，テキスト画面に半角文字で表示する方法

【キー入力】

(c-1) 16進数（4桁）で入力する方法

(c-2) 日本語の漢字変換と同様に，平仮名で読みを入力して変換する方法

(c-3) 半角カタカナのキーを用いて半角カタカナのコードを入力する方法

(c-4) 英語文字でまとめて入力しそのあと一括して変換する方法

(c-5) 入力モードを切り替えることにより，第3国語の各文字がキー入力1回で可能であり，しかも1文字入力ごとに表示もされる方法

以下，それぞれの方法の問題点についての分析を述べる。

(a-1) ユーザ文字定義の2バイトコードに割り当てる方法

第3国語の各文字には，あらかじめ許されている2バイトのコードが割り当てられる。1バイトのコードを割り当てる方法と比べると，表示した場合同じ文字数となる場合でもファイルの大きさは2倍になる。したがって，保存するディスク上での占有領域も2倍となる。小さい領域に収まることが望ましいけれども，最近のディスクの大容量化の状況を考えれば，このことはさして問題ではなくなりつつある。

しかし，扱えるデータが64 Kバイトあるいは利用可能な主記憶容量などで制限されているソフトウエアを用いるときには問題となる。われわれは今後の言語研究で必要とされるソフトウエアをすべて開発するのではなく，可能な限り市販のソフトウエアも用いることを想定している。したがって，第3国文字すべてに2バイトコードを割り当てるのは多少問題がある。

(a-2) ワープロソフト独自の2バイトコード体系を用いる方法

(a-1)と同様な問題がある。ワープロソフトではイタリックなど複数の文字フォント，強調文字や下線文字などの文字修飾などのために，1文字あたり4バイト以上も用いているシステムもある。また，そのような文字修飾の機能が一切無い場合でも2バイトコードを使用している場合が多い。

文字コードの選択と密接に関連する問題として，ワープロソフト以外のソフトウエアの使用時における表示の問題がある。独自のコードの場合，そのワープロソフト以外では，外字登録された文字コードの表示が正しくなされない場合が多い。たとえばファイルの内容を表示させるMS-DOSのコマンドtypeなどは利用できない。type程度であれば，代用のプログラムを作成する事により解決できるが，データベースなどの他の多くの市販のソフトウエアの場合には代用プログラムの開発は実際には不可能であろう。

(a-3) 半角カタカナの1バイトコードを割り当てる方法

この方法では，上記の2つの方法で指摘した扱える文字数についての問題点は排除できる。ただし，半角カタカナのコードは16進数でA0からDFまでの64文字であり，第3国語の文字だけであればほとんどの場合十分であるが，第3国語の特殊記号などを含めると不足する場合も予想される。

したがって，よく用いる第3国語の文字には1バイトコードを，それ以外には2バイトコードを用いることができればよいが，そのような方法をとれるシステムは現在のところ存在していない。

(b-1) MS-DOSのユーザ定義文字の機能を用いて全角文字で表示する方法

CRTの1画面に表示できる文字数は半角文字の場合に比べて半分になる。論文・文章などを考えながら入力する場合には，できるだけ広範囲のテキストが見えていることが重要であり，したがって，全角文字での表示は問題がある。

なお，この方法では，各種ユーティリティ・ソフトウエアを利用するときにも全角文字ではあるが，文字表示は問題なくおこなえる。

(b-2) 独自の外字登録機能で，グラフィック画面に半角文字で表示する方法

この外字登録はワープロ以外では働かない場合が多く，(a-2)で述べたように，ワープロソフト以外の各種ユーティリティでの文字表示が問題となる。

(b-3) CRT表示をトラップしグラフィック画面に半角文字で表示する方法

半角文字として表示するため1画面に表示できる文字数での問題はないが，この方法が採用されているシステムでは，割り当てているコードが半角カタカナのコードの部分であり，表示文字数が64文字に制限されていることが問題である。

なお，最近のソフトウエアの中には，画面表示の高速性・多彩さを実現するため，MS-DOSのCRT表示のソフトウエア割り込みを利用せずに，直接テキストVRAMへ文字コードを書き込むものもあるが，そのようなソフトウエア

に対しては，この方法でのトラップは不可能であるという問題もある。

(b-4)定義機能と表示トラップでテキスト画面に半角文字で表示する方法

問題点は(b-3)と同様である。

(c-1)　16進数（4桁）で入力する方法

第3国語の1文字を入力するのに最低4回（システムによっては，1文字ごとに16進数入力モードにしなければならず，その場合には4回以上)のキー入力が必要であり，大部分が第3国文字である文章の入力には適さない。

(c-2)　日本語の漢字変換と同様に，平仮名で読みを入力して変換する方法

この方法の場合も，(c-1)と同様に第3国文字の入力において複数回のキー入力が必要とされることが問題となる。

(c-3)　半角カタカナのキーを用いて半角カタカナのコードを入力する方法

通常は，「カナ・キー」を押してロックすればカタカナ入力モードとなり，再度カナ・キーを押し，ロックを解除するまでこのモードは保たれる。そして，カタカナ入力モードでは第3国語の文字1文字の入力は1回のキー入力となるので，(c-1)や(c-2)のような問題はない。

ごく自然な文字コードの割り当ては，英語の文字のように大文字のコードに16進数で20を加えたものが小文字のコードとなるものであろう。そして，キー入力においては，英文字と同様に，シフトキーを用いることにより，大文字と小文字は同一のキーで入力できることが必要であろう。しかし，カナ・キーをロックしたカタカナ入力モードではシフトキーを用いても1つしか文字コードを入力することができないキーもある。したがって，すべての文字についてシフトキーを用いることにより大文字と小文字を同一のキーで入力することは不可能である。また，たとえシフトキーが働くキーに対してでも，文字コードの割り当て方を不自然なものにしなければ，同一のキーでの大文字と小文字の入

力は可能とはならない。

また，英語だけでなくロシア語などの第3国語のうちには，以前からタイプライターが利用されている言語もあり，それらの言語の研究者はタイプライターのキー配置に慣れていると思われるが，その配置と半角カタカナコードでの入力の配置，さらに，自然な文字コードの割当をすべて満足することは不可能である。

(c-4) 英文字でまとめて入力しそのあと一括して変換する方法

この方法の場合には，英語の文字と第3国語の文字の対応を完全に記憶しておく必要がある。対応が不確かな場合には各文字の入力毎に確認できず，スペルミスが生じやすいという問題がある。

(c-5) 入力モードを切り替え第3国語の各文字がキー入力1回で可能な方法

大部分が第3国文字である文章の入力には，この方法が最も適していると思われる。ただし，この方式がとられているのは独自のコード体系を持ったワープロソフトの場合であり，(a-2)で述べたのと同じような問題がある。

2.2 第3国語の取扱いが可能なデータベース・ソフトウエアの現状

市販のほとんどのデータベース・ソフトウエア（以後データベース・ソフト）では，日本語と英語の表示は問題がないが，第3国文字および特殊文字などが扱えるものは，我々の知る限り1つだけ——「日本オフィス機器」の『マイクロコスモス』[11]——しか存在しない。しかし，そのソフトウエアには次のような問題がある。

(1) そのデータベース・ソフトは文字コード体系は(a-3)，表示は(b-3)，入力は(c-3)の方法をとっている。したがって，2.1で述べたように，入力効率の悪さ，半角カタカナ以外の2バイト文字コードを用いなければならない場合には，CRT表示での問題がある。

(2) カード型・対語型のシステムであり，データベースを操作する言語が備

わっていない。このことは，時間を要する複雑な処理などであっても，すべてユーザが対話的に処理しなければならず，非常に効率が悪い。

(3) このソフトウエアについてだけでなく，一般のパソコンのデータベース・ソフトについても該当することではあるが，各レコードに同様なレベルのキーワードを複数個持たせる必要のある場合に効率のよい検索ができないという問題がある。

たとえば，各書籍に複数のキーワードが設定される書籍データベースを考えてみる。複数のキーワードのため，通常は複数のキーフィールド（たとえば，kf1, kf2, ……kf10）を設ける。このとき，各キーワード（たとえば，「言語」がどの書籍に対しても，常に決まったフィールドに入れられるとは限らないことが問題となる。つまり，キーワード「言語」が書籍1の場合にはkf1に，書籍2の場合にはkf2に，……と入れられることもある。したがって，「言語」での検索にはkf1 …… kf10すべてについて調べなければならない。まして，OR条件やAND条件を組み合わせた複雑な検索の場合には，迅速な検索ができないことが多い。

もちろん，データベース・ソフトの中には，検索，言語の有無などにおいて，上記のような問題点の無いものもある。しかし，それらは，第3国語の取扱いにおいて問題があるのが現状である。また，そのようなデータベース・ソフトの場合には，簡単な処理に対しても，そのデータベース操作言語でプログラムを作成しなければならず，初心者にはその使用は若干難しいかもしれない。

2.3 言語学の研究補助に用いられるユーティリティについて

言語学研究用に必要なユーティリティを十分に把握していないのが現状である。そのため，必要に応じて適当な市販あるいはPDSなどのソフトウエアを探しているが，入力・表示・機能の3点を満足する適切なものは見つかっていない。また，現在のところロシア文字などを扱えるワープロで，われわれの目的にあうユーティリティは調査したところでは存在していないし，また開発することも不可能ではないにしても非常に困難である。

一方，われわれが対象としているパソコン(NEC PC-9801)では，入力と表示の問題が解決すれば，ユーティリティの開発はさして困難ではなくなるし，たとえばunixなどの上での文書ファイルに対する文字列を扱うソフトウエアとして有名なawkなどは利用可能となり，強力なユーティリティとなることが期待できる。

III 言語学研究のためのシステムの基本仕様

以上の分析から，既存の市販ソフトウエアを組み合わせただけでは，満足できるシステムは構成できないと判断した。そこで，キー入力，画面表示，テキスト・ファイル作成のためのエディタ，ユーティリティ，データベース機能を有した，第3国語のための言語学研究——最初はロシア語に関する研究・教育，およびブルガリア語の活用辞書作成等を目的とする——のための環境を与えるトータルシステムを開発することにし，まずその基本的な仕様を次のように定めた。

(1) 文字コード

- 第3国語の文字には1バイトのコードを用いる。カタカナを必要とする場合は全角文字を用いることにし，半角カタカナのコードを割り当てる。
- 今後の処理を考え，大文字と小文字のコードの対応は自然なものとする。
- 特殊文字などは，2バイトのコードでもよい。

(2) 表示

- 表示は半角文字でおこなう。
- 表示方式は，表示プログラムをトラップしてテキスト画面あるいはグラフィック画面に表示する方式とし，われわれ自身の開発する特定のソフトウエアだけでなく，市販のユーティリティ，データベース・ソフトなどでも，可能な限り第3国語が表示されるように配慮する。
- イタリック，強調文字，下線文字などの文字修飾は，最終的な印刷で表現されればよい。CRT画面では，文字修飾の種類とその開始と終了の範囲がわかる程度の表示でもよい。

(3) 入力

・第3国語の文字入力のモードを設けることにより，そのモード中では1文字1キー入力とする。読み入力後に変換キーを用いる方式は避ける。

・シフトキーを用いることにより，大文字と小文字は同じキーで入力できるようにする。

(4) データベース機能

・第3国語の表示が可能であるものとする。

・必要とされる程度のさまざまな検索機能が備わっているものとする。

・データベースを操作するためのプログラム言語が備わっているものが望ましい。

・ただし，そのような市販のデータベース・ソフトが存在しない場合には，ISAM ライブラリなどを用いて開発する。そのときには，必要なデータベース操作に対する十分なコマンドを備えることができれば，操作のための言語を備えなくてもよい。

(5) ユーティリティ

・われわれ自身もどのような機能のユーティリティが必要かつ十分なのかは把握していないこともあり，必要とされる機能が判明するごとに分析・設計・開発・改良をおこなうことにする。

・なお，現在までに開発が必要で有用であろうと考えた言語学研究のためのユーティリティは，次のようなものである（これらについては「Ⅷ言語学研究におけるユーティリティの開発について」で説明する）。

(a) 作品などのファイル中の単語の出現調査のユーティリティ

(b) 作品などのファイル中の熟語の検索ユーティリティ

ただし，これらの仕様を満足するトータルシステムの開発には時間を要するので，さしあたり我慢すれば利用できるものは市販のものあるいは PDS を利用することにし，是非とも必要なものから開発をおこなうことにした。

まず開発の必要なものは，第3国語の入力・表示の可能なエディタ（開発ソフト名 RMACS. EXE) と印刷ソフトウエア (RLIST. EXE)，市販品や PDS な

どが利用できない辞書作成のための各種ユーティリティ，第3国文字のキー入力プログラム (RUS. SYS) などである。なお，半角カタカナのコードを第3国語の文字コードに用いることに決定したため，表示プログラムをトラップしグラフィック画面に第3国文字を表示することが可能な市販のソフトウエア「bits」を利用することができる。ただし，2バイトコードを用いる特殊文字などでは表示が乱れることがある。なお，「bits」の問題点であったキー入力は，開発する RUS. SYS で解決が図れる。また，データベースについては，まだ必要な機能が確定しないこともあり，当面は「bits」が利用できて第3国文字の扱いの可能な唯一のものである「マイクロ・コスモス」を試験的に用いて機能の検討を重ねることにした。

パソコンを用いてどのような処理を行うにしても，必要なテキストの入力のためには，エディタは重要な道具である。しかも，紙・ノートおよび鉛筆代わりにパソコンを利用し，気がついたことをメモしておくことが非常に役立つことも多く，そのためにも，使い勝手のよいエディタは重要である。しかし，市販のワープロやエディタでは，そのほとんどのものが画面表示の高速化を図るため，CRT への表示は直接テキスト VRAM へ書き込む方式をとっている。したがって，CRT 表示プログラムをトラップする方式が利用できない。そのため，エディタの開発を最初に行うことにした。

日本文字と英文字を扱うことのできる GNU-Emacs のコマンド体系のサブセットを有したエディタを開発した経験もあり，それに手を加えることにより同じコマンド体系を有して第3国語の文字も扱えるものを作成した。このエディタ (RMACS. EXE) の設計思想やコマンド体系などは，[5][6][7]を参照されたい。なお，エディタでの第3国文字のコードの入力には，開発したキー入力プログラムを利用している。表示には「bits」を用いることも可能ではあるが，画面表示の高速化を図るためと，エディタ (RMACS. EXE) だけでも使用したい人のために「bits」の著作権にふれないことを考慮して，現段階ではエディタ独自でグラフィック画面に第3国文字を表示するようにしている。「bits」に代わるプログラムを開発した時点で，エディタでもその機能を用いるようにす

る予定である。

なお，「bits」とは別に，エディタで独自に第3国文字の表示を行うので，エディタのためのCRT表示用フォント・パターンを作成しなければならない。そのために，フォント・パターン作成用プログラム(FONTD.EXE)を作成した。このプログラムはロシア文字のプリンタ出力で必要とされるフォントの作成にも利用できるように汎用性を持たせた設計としている。

ところで，一時よく議論されたことでもあるが，論文などの文章を作成するのにワープロ（ソフト）がよいのか，テキスト・エディタがよいのかということがある。われわれ自身は，次のように考えている。

ワープロは強調文字，下線文字，網掛け文字などの多彩な文字修飾の出力を想定しており，しかもそのような文字修飾の設定がCRT上でも容易に確認できるWYSWYG (What You See is What You Get) 方式を採用している。

これに対して，テキスト・エディタはそもそもプログラムやデータなどのテキスト・ファイルを作成することを想定しており，多彩な文字修飾などには重点はおかれていない。しかし，論文などの文章もテキスト・ファイルとして作成できるし，さらに印刷プログラムなどを工夫すれば，かなりの程度までの文字修飾も可能となる。ただし，WYSWIG方式まではサポートされていない。

われわれの経験からは，カーソルの移動の速さ，画面表示の速さ，編集のためのコマンド体系などから，文字修飾の無い文章を作成する場合には，テキスト・エディタに分類されているものの方が使用しやすいと感じている。また，WYSWYG方式ではないが，数学論文の作成用ソフトウエアとして有名なTEX［1］［2］のように，印刷された状態がかなり正確に，印刷以前にCRT画面上で確認できれば十分であると考えた。そして，現時点では，おおまかな印刷された状態がわかる程度にファイルへ出力を行う機能を印刷プログラムに設けることにした。

IV　文字コードの割り当て

われわれの当面の対象であるロシア文字およびその特殊文字に対する文字

	A	B	C	D
0	А	Р	а	р
1	Б	С	б	с
2	В	Т	в	т
3	Г	У	г	у
4	Д	Ф	д	ф
5	Е	Х	е	х
6	Ж	Ц	ж	ц
7	З	Ч	з	ч
8	И	Ш	и	ш
9	Й	Щ	й	щ
A	К	Ъ	к	ъ
B	Л	Ы	л	ы
C	М	Ь	м	ь
D	Н	Э	н	э
E	О	Ю	о	ю
F	П	Я	п	я

図1　ロシア文字のコード割り当て
（1バイトコード）

	FA2
0	Ё
1	ё
2	≪
3	≫
4	”
5	“
6	—
7	№

図2　ロシア文字のコード割り当て
（FA2x の型の2バイトコード）

コードは次のように決定した。ロシア文字64個に対しては，半角カタカナのコードを用い，大文字と小文字のコードの対応の自然なものとする。そのコード割り当てを図1に示す。特殊文字などのコードは，図2のように，Shift-JISコードの拡張用エリア（第1バイトが16進数でF0からFDまでの2バイトコード）を用いることにする（2）（3）。ただし，このコードエリアに割り付けることのできる文字数は最大256×14個（このうち200×14個程度を用いるのが普通）あり，現在の特殊文字数6より圧倒的に多い。現時点で必要としている特殊文字だけならこのエリアの第1バイトのみを利用しても十分であるが，そうしなかった理由は，今後の特殊文字あるいは発音記号(130文字種以上)など将来の拡張を考慮した結果である。

（2）　試作の段階では，6個の特殊文字にはそれぞれ16進数でF0からF5までの1バイトコードを割り当てていた。

（3）　今回の割当で第1バイトにFEとFDを割り当てない理由は，MS-DOSではこれらのコードは特別に扱われるという制限があるからである。

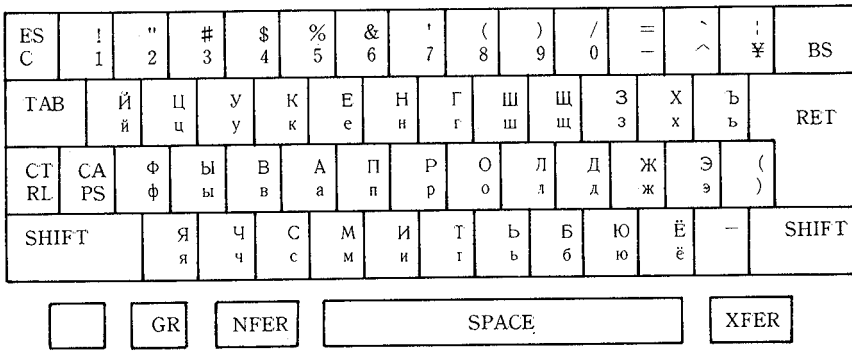

図3　ロシア文字のキー配置（通常キー）

HOME CLR	HELP	№ —	≫ ≪
7	8	9	“ ”
4	5	6	? +
1	2	3	;
0	,	.	RET

図4　ロシア文字のキー配置（テン・キー）

V　キーボード上のキー配置

ロシア文字についてはIBMのロシア語用タイプライタを参考に，図3のように配置した。英文字の特殊文字はパソコンのキートップの表示とできるだけ合わせて，ロシア文字の特殊文字は主としてテンキーの部分で入力することとし，図4のようなキー配置とした。

なお，入力モードの切り替えは，コントロール・キーを押したままグラフ・キーを押すことにより英文字入力モードとロシア文字入力モードがトグル・スイッチのように切り替わる方式とする。どちらの入力モードであるかは，カー

ソルの形状で判断できるようにする。通常の入力モードのときはリバース表示，ロシア文字入力モードのときは下線表示とする。なお，日本文字の入力モードの切り替えについては，使用する日本語フロント・エンド・プロセッサに依存するが，多くの場合は，英文字入力モードでコントロール・キーと XFER キーを押す方法がとられている。

VI ロシア文字などの入力の実現方法

キー入力に対する処理としては，コントロール・キーとグラフ・キーが押された場合に入力モードを切り替える処理と，ロシア文字入力モードで入力された英文字コードをロシア文字コードへ変換する処理からなる。

われわれがシステム実現の対象としているパソコン (NEC PC-9801) および

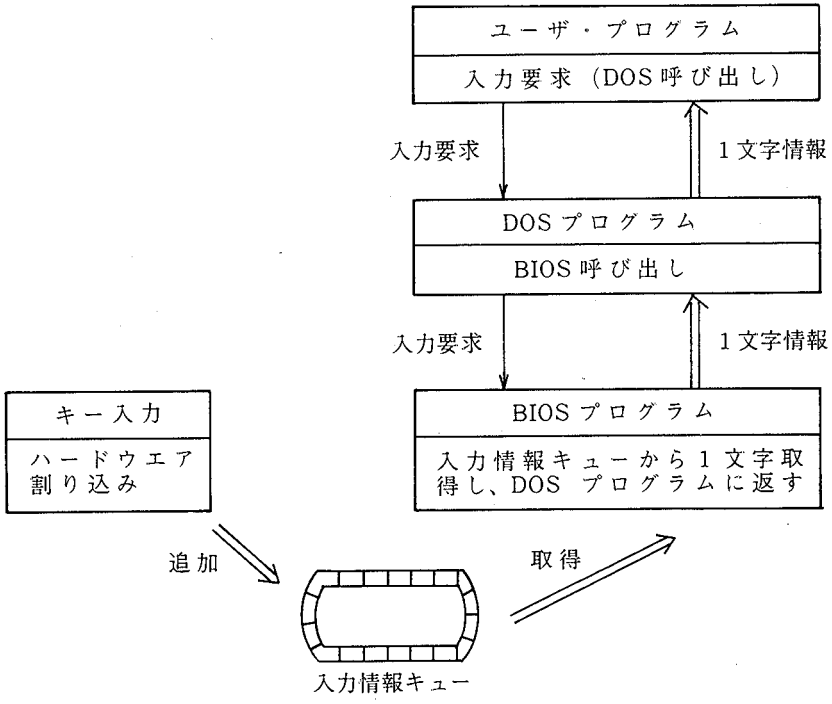

図5 キー入力の処理

MS-DOS でのキーボード入力の処理は次のように行われている（図5）。

① キー入力があると，パソコンのハードウエア割り込み（ベクタ番号9）が発生し，どのキーが押されたかという情報が，割り込み処理プログラムにより分析され入力キューの最後につけ加えられる。

② プログラムからの入力要求は，MS-DOS の入力用 DOS 機能の呼び出しを用いて行われる。

③ その DOS 機能は，さらにコンソール入力用のデバイス・ドライバを呼び出す。

④ さらにデバイス・ドライバはソフトウエア割り込み（ベクタ番号 18 H）で BIOS 機能を呼び出す。

⑤ BIOS では，①の処理でキューに入れられているキー情報を先頭から取り出してデバイス・ドライバに渡す。そして③，②を逆に戻り最終的にはプログラムに入力文字コードが渡される。

ただし，コントロール・キーとグラフ・キーの入力は文字コードを返さないので，DOS 機能の呼び出しでは認識できない。したがって目的とする処理は，①のキー入力の割り込み，③のデバイス・ドライバ，あるいは④の BIOS 呼び出しのいずれかの段階をトラップすることにより可能となる。

(1) キー入力要求の BIOS 呼出のトラップ

BIOS 呼び出しをトラップするには，MS-DOS が立ち上がったときにセットされている割り込みベクトルの 18 番（0000：0060 番地）の 4 バイトの内容をトラップ処理用に作成したプログラムの番地に変更することにより可能となる（図6）。

ただし，BISO 呼び出しに完全に置き代わるプログラムの開発は大変であるので，図7のフローチャートのように元の BIOS を利用する方法を採用した。この方法は，プログラム作成作業を軽減し，しかもプログラム領域を小さくできるという長所がある。

この方法で実現したものをしばらく利用していたが，ワープロソフトとして

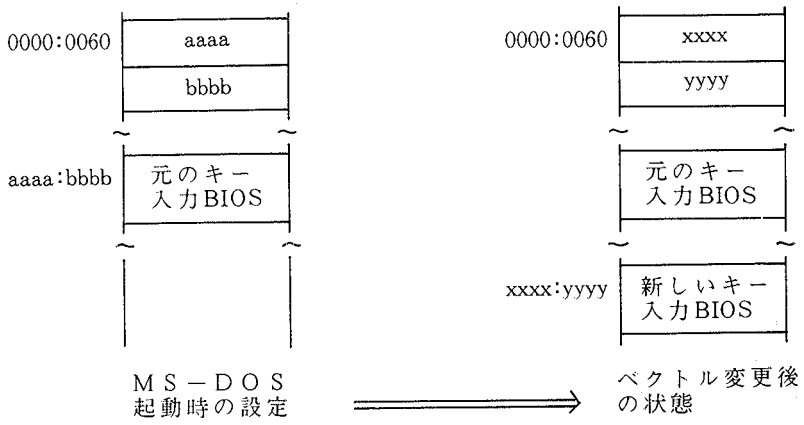

図6　BIOS呼び出しのトラップ（割り込みベクトルの変更）

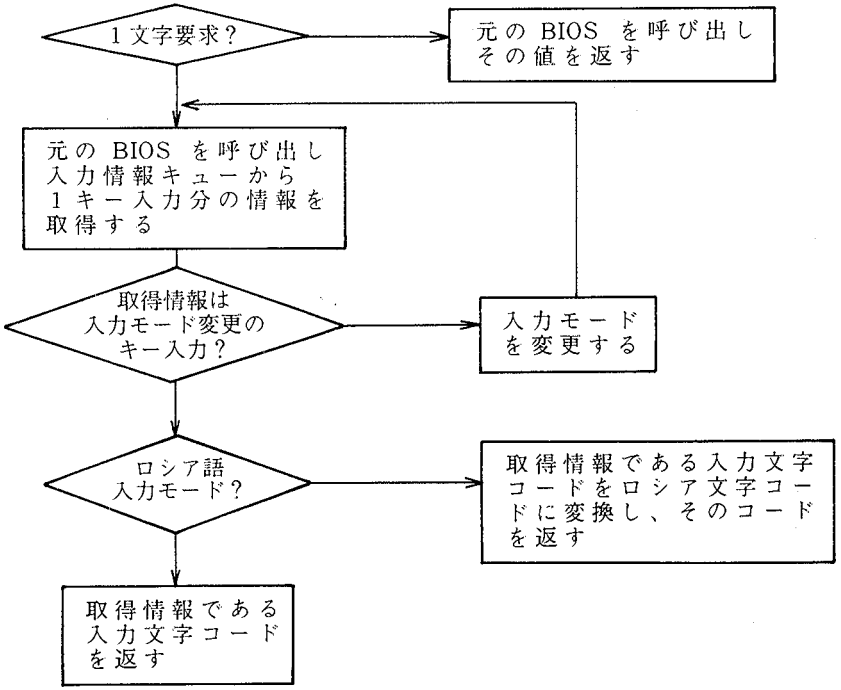

図7　開発したBIOSの処理方式

有名な『一太郎』付属の日本語フロント・エンド・プロセッサATOKと併用した場合に，ロシア文字入力モードでコントロール・キーとXFERキーを押して日本語変換モードにしたあと，ロシア文字入力モードを解除するという順序でキー操作を行うと暴走するという問題が生じ，簡単には原因の究明ができなかった。「バックス」の『VJE』など他のフロント・エンド・プロセッサでは問題は生じていないが，『一太郎』との関連で，現在広く使用されているATOKを無視することもできないため，次に述べるデバイス・ドライバをトラップする方法をとることにした。

(2) キー入力要求のデバイス・ドライバでのトラップ

トラップするデバイス・ドライバは，新たにコンソール入出力用のデバイス・ドライバを作成し，CONFIG.SYSファイル中で

DEVICE＝作成したデバイス・ドライバ・ファイル名

と指定することにより，登録することができるし，またMS-DOSに付属のソフトウエアADDDERV.EXEを用いて登録することも可能である。

この方法の場合も完全なデバイス・ドライバを作成するのではなく，開発作業の軽減とプログラム領域の小型化のために，既にインストールされているキーボード入力用のデバイス・ドライバを利用し，モード変更とコード変換を行う部分のみを開発する方法をとる。なお，処理方式は，BIOSをトラップする方式の図7での処理における「元のBIOS」の部分を「元のデバイス・ドライバ」とすることにより得られるものと同様な処理となる。

Ⅶ ロシア文字などの印刷

当然のことながら，エディタで作成したテキスト・ファイルなどを印刷する機能も実現しなければならない。一般に，パソコンに接続できるプリンタでは，全角のロシア文字であれば，日本文字や英文字などのように文字コードを送って印字することも可能である。しかし，全角文字では記号として用いる場合にはともかく，文章を印字するには大きすぎて不適切である。また，通常のプリ

ンタではロシア語全角のイタリック文字などは印字できない。しかも，半角ロシア文字には半角カタカナのコードを流用しているので，文字コードをプリンタに送ると半角カタカナが印字される。また，そもそも，われわれが対象としているパソコン(NEC PC-9801)用のほとんどのプリンタは半角ロシア文字のフォント・パターンすら有していない。

したがって，印字させるにはプリンタに対してフォント・パターンを送って図形として印字させなければならない。そのため作成した印字プログラム中にフォント・パターンをデータとして組み込んでいる。そのためのフォント・パターンの作成には，先に述べたプログラム FONTD.EXE を用いることにしている。

ワープロで可能である多彩な文字修飾を，開発する印刷プログラムですべて実現するつもりはないが，日本文字，英文字，ロシア文字について拡大，強調，下線程度の文字修飾は，NEC の PC-PR 201 系統のプリンタに対しては，つぎのようなプリンタに対する命令をテキスト・ファイル中に埋め込んでおくことにより可能である。これ以外に罫線機能が実現されれば，実際上では十分であろうと思われる。

	開始		終了	
拡大：	ESC e 22	拡大する文字列	ESC e 11	など
強調：	ESC ！	強調する文字列	ESC ”	
下線：	ESC X	下線を付ける文字列	ESC Y	

- ESC はエスケープのコード（16 進数で1B）を表す。
- 開始，終了の指定の間の空白（ESC と e の間の空白等）はファイル中では空けない。

また，ロシア語の研究者，あるいはロシア語で文章を書く人にとっては，ロシア文字のイタリック体での出力も必要である。また，英文字についてもイタリック体が利用できることが望ましい。そこで，先のフォント・パターン作成

プログラムFONTD.EXEを用いて英文字とロシア文字のイタリック体フォント・パターンの作成をおこない，それらを印刷プログラム中にデータとして組み込んでいる。そしてテキスト中の「コントロールＸ１（以後^X1のように記述)」と「^XØ」との間の英文字あるいはロシア文字はすべて，イタリック体での印字となる。つまり，「^X1」がイタリック体印字の開始，「^XØ」が終了を意味する。イタリック体の文字に対しても，拡大，強調，下線の文字修飾は可能である。図8(1)(2)は本システムでのロシア文字，日本文字，英文字の混じった印字例である。図8(2)ではロシア文字に対する拡大，下線，イタリック体などの文字修飾もなされている。

なお，出力プログラムはNECのPC-PR201シリーズを対象プリンタとして

現代ロシア語の連辞動詞の過去形である(был, было, была, были)は古ロシア語(древне-русский язык)では分詞形として用いられていたものである。ちなみに Lausitz という地名はロシア語の луг，「草原」の意であり，この地方の草原に住むスラブ人のことをロシア人達がラウジッツ人(лужичанин, лужицкий «形容詞形»)と呼んだのもこのためであった。

図8(1) 印字例：ロシア文字＋日本文字＋英文字

ТРИ ГОДА

Чехов, А. П.

Было еще темно, но кое-где в домах уже засветились огни и в конце улицы из-за казармы стала подниматься бледная луна. Лаптев сидел у ворот на лавочке и ждал, когда кончится *всенощная* в церкви Петра и Павла. Он рассчитывал, что Юлия Сергеевна, возвращясь от *всенощной*, будет проходить мимо, и тогда он заговорит с ней и, быть может, проведет с ней весь вечер.

図8(2) 印字例：ロシア文字

開発しているので，他の機種では必ずしも正常に機能するとは限らない。ただし，上記プリンタの互換モードを持っている機種——たとえばNEC NMシリーズ，キャノンのBJ-130など——でも，一応は動作するようである。プリンタについては，ページプリンタも含めてあまりにも機種が多いため，印刷プログラムをすべてのプリンタに適用させることは現時点では考えていない。しかし，高度な印字品質が得られることから，対象機種として代表的なページプリンタを含めることは考慮中である。

Ⅷ 言語学研究におけるユーティリティの開発

言語学研究の分野でパソコンを利用する方法の検討を始めたばかりであるため，まだ，どのような機能・ユーティリティが必要・有用なのか十分には把握していない。したがって，研究・辞書作成を進めるにしたがって，必要かつ有用であると思われたものを開発してきている。現在までに開発を行っているものは，つぎの2点である。これらは，研究・教育のために単語帳や熟語帳を作成したいという話に対して，プログラム開発で有効に用いられている，クロス・リファレンス作成プログラム，および文字列パターンの検索プログラム(grep)をもうまく利用できるのではないかということで開発したものであるが，実際に使用してみると非常に有用かつ効果的に利用されている。

現在のユーティリティは，このような経緯で開発に着手し，徐々に機能を追加する形で発展されたものである。したがって，現段階で最終的に確定した仕様とは，認識しておらず，さらに利用し易いものへと改良を加えるつもりである。

(1) 作品などのファイル中の単語の出現調査のユーティリティ

以下に示す仕様のOCCUR.EXE，MERGE.EXE，FMT.EXE，UNFMT.EXE，HSORT.EXEの5つのユーティリティからなる。なお，以下の【使用法】の記述で［と］で囲まれた部分は省略可能である。また，小文字のn，m，kなどは10進数を表し，「-」の直後のP，L，O，Wなどは大文字・小文字どち

らでもよい。

OCCUR.EXE

【使用法】

A > OCCUR text_file dest_file [-Pn] [-Lm] [Ox]

text_file	：テキストが入力されているファイル。
dest_file	：結果の出力ファイル。
-Pn	：開始ページ指定
-Lm	：開始行指定
-Ox	：単語を辞書式順序に並べるか(x＝Y), テキストでの出現順にするか(x＝N)。

【説明】

text_file 中に出現する単語をすべて切り出して，次に示すような形式でその出現しているページと行の情報を付加して dest_file に出力する。

単語1 ： page11-line11, page12-line12, page13-line13

単語2 ： page21-line21, page22-line22, page23-line23

したがって，作品あるいはテキストなどの文章を入力しておくファイル text_file では，行の区切りには CR を，ページの区切りにはコントロール L を用いて原文通り入力しておくことが望ましい。

なお，作品などが大きくいくつかのファイルに分割して入力している場合，各ファイルの最初が原文において何ページ(n)の何行(m)であるかを-Pn および-Lm で指定することができる。省略したときは，n，mとも1が指定されたものとする。

最後のオプション-Ox で，出力ファイル中での単語の並びを辞書式順序とするか，あるいは原文での出現順にするかの指定ができる。省略された場合は x＝Y の指定と同じである。

MERGE.EXE：

【使用法】

A > MERGE source_file1 source_file2 dest_file

source_file1 : OCCUR.EXE で作成されたのと同じ形式のファイルであり，辞書式順序のもの。

source_file2 : source_file1 と同様。

dest_file : source_file1 と source_file2 を辞書式順序にまとめたファイル。

【説明】

大きな作品の場合，いくつかのファイルに分割して入力する方が取扱いが容易である。そのような場合に，ファイル別に OCCUR.EXE を適用して作成したものを，MERGE.EXEを用いて辞書式順序でまとめる。

FMT.EXE:

UNFMT.EXE:

【使用法】

A > FMT source_file dest_file [-Wn]

source_file : OCCUR.EXE で作成されたのと同じ形式のファイル。

dest_file : 各単語に対する出現位置の並びを適当な位置で改行で区切ったファイル（説明参照）。

-Wn : 1行の（半角文字での）文字数を n で指定する。省略時は 100 が指定されたとする。

A > UNFMT source_file dest_file

source_file : FMT.EXE で作成されたのと同じ形式のファイル。

dest_file　　：OCCUR.EXE で作成されたのと同じ形式のファイル。

【説明】

OCCUR.EXE で作成されるファイルでは，1行のサイズは無限としている。しかし，印刷時に使用する用紙によっては当然サイズは有限であり，サイズを変更する必要がある。FMT.EXE はそのためのユーティリティである。たとえば，CRT 画面上で折り返して表示される長い行

単語1：page11-line11, page12-line12, page13-line13, page14-line14, page13-line13, page13-line13,page13-line13

は，FMT.EXE を用いることにより，

単語1：page11-line11, page12-line12, page13-line13,
：page14-line14, page13-line13, page13-line13,
：page13-line13

のように，改行コードが入れられて変更される。

UNFMT.EXE は，EMT.EXE の逆を行う。いったん FMT.EXE で作成したファイルと，OCCUR.EXE で作成したファイルに対して MERGE.EXE を適用する場合などに用いる。

HSORT.EXE：

【使用法】

A > HSORT source_file dest_file

source_file　　：OCCUR.EXE で作成されたのと同じ形式のファイル。単語の並びは辞書式順序でなくてもよい。

dest_file　　：OCCUR.EXE で作成されたのと同じ形式のファイル。単語の並びは辞書式順序となる。

【説明】

OCCUR.EXE で，テキスト中での出現順でファイルを作成し，さらにエディタなどで情報をつけ加えた後，そのファイルを辞書式順序に並べ変えるためのユーティリティである。

(2) 作品などのファイル中の熟語の検索ユーティリティ

まず，このユーティリティ MLGREP.EXE を用いれば，どのようなことが可能であるのかを英語の場合を例に説明する（4）。

例1：

まず，あるファイル中の「look for」という熟語が使われている文（部分）をすべて検索することを考えてみよう。通常のエディタは単なる文字列検索の機能しか有していないため，つぎのような問題がある。

(1) look と for の間の空白の個数が同じものしか探索できない。あるところでは1つ，別のところでは2つの空白があればどちらかしか探索できない。

(2) 原文などの関係から look で改行し，次の行の先頭に for が位置する場合には，間を空白にしたのでは探索できない。

本ユーティリティを次のように用いれば，このような場合の文字列検索も可能である。ただし，目的としている以外の「look for」も見つかる場合もあり，探索し，出力されたものを人間が確認しなければならない。しかし，look あるいは for を単独に検索し，その結果を確認する場合に比べるとはるかに楽であろうと思われる。このことは例2についても同様である。

A > MLGREP look[ ]+for ファイル名

例2：

「mix A with B」のように熟語を構成する単語の間に他の単語が入るような

---

(4) 本ユーティリティ MLGREP.EXE は，本田が開発した1行中で指定されたパターン1個を検索する言語Cで記述したプログラム hgrep.c を，本学経済学部管理科学科助手の吉岡珠実氏が改良し，複数行中での複数パターンの検索機能を追加したものである。

場合は例1での問題もあるが，それ以上に通常の探索機能では不可能である。しかし，本ユーティリティでは次のようにして可能である。

A > MLGREP mix * with ファイル名

あるいは単に

A > MLGREP mix with ファイル名

MLGREP.EXE:

【使用法】

A > MLGREP [-n] [-Pm] [-Lk] パターン1 … パターンk source_file

パターン	：正規表現の文字列
source_file	：テキストが入力されているファイル。
-n	：熟語の検索対象行数。
-Pm	：開始ページ指定。
-Lk	：開始行指定。

【説明】

source_file 中で，指定されたパターンがその順序で出現する部分をCRT 画面に表示する。この表示はMS-DOS のリディレクションの機能を用いてファイルに取り込むことも可能である。パターンの記述には正規表現が利用できる。なお，正規表現については，「X　付録」として説明する。

source_file には，MS-DOS のファイル指定で利用可能な＊や？などを用いて複数のファイルを指定することもできる。

検索対象となる行数をn（$1 \leqq n \leqq 5$）で指定することができる。省略時は$n = 5$と解釈される。

大きな作品の場合，いくつかのファイルに分割して入力されている場合，ファイル別に，最初の部分以外のファイルに対しては，そのファイルの先頭が元の作品の何ページの何行であるかを指定する必要がある。その指定は-Pm および-Lk で行う。

IX さいごに

このように，ロシア語を含む第3国語の扱えるエディタに始まり，言語学研究のための環境を整えるユーティリティを開発してきたが，利用すればするほど，新たな機能の要請が生じ，機能追加としては以下のものを考慮中である。

(1) 辞書作成のために発音記号も扱えるようにする。

(2) 特に研究のためであるが，古代ロシア語も扱えるようにする。

(3) ロシア語の場合，テキスト中の単語にアクセントをつけることがある。したがって，アクセント付きロシア文字も扱えるようにする。

このように，辞書作成とか教育用の資料作成などのためには，今後ともユーティリティの一層の検討・開発が必要である。また，データベース機能についても検討が是非とも必要となるであろう。われわれとしては，そのような要請に答えていくつもりである。

X 付録——正規表現について

まず，正規表現に使用できる文字の集合（アルファベット）Σは通常の文章やプログラムが記述できるような文字の集合として，次の(a)〜(f)で定められるものとする。

(a) タブ，空白。なお，タブ以外のコントロール文字(16進数で00から1Fの文字) はΣには含まれないとする。

(b) ”等の特殊文字。ただし，以下の文字はあとで説明するように正規表現中では特別な意味を持って使用されるので，アルファベットには含まれないものとする。これらの文字を，特別の意味を持たない文字として正規表現中に使用するには，(e)または(f)による文字に該当するようにして使用すればよい。

^ $! * ? = + ¥ [ ]

(c) 数字，英大文字，英小文字，カタカナ等の1バイト文字。

(d) 2バイトのシフトJISの日本語文字。

(e) ¥xy の形のもの。ただし，x と y は16進数である。(見かけは計3文字となるが，Σに含まれる1文字として考える。これはそのコードが16進数で xy の文字をあらわす。ただし，そのコードはタブ以外の16進数で1F以下の数であってはならない。)

(f) ¥c の形のもの。ただし，c および c に続く文字 d は2文字の16進数とは考えられない文字である。

(見かけは計2文字となるが，Σに含まれる1文字として考える。これは文字 c をあらわす。)

アルファベットΣの上の正規表現とは，以下の規則から作ることのできる表現（文字列）のことである。

(a) Σ内の各文字 c に対して，その文字だけからなる表現 c は正規表現である。

(b) および？だけからなる表現は正規表現である。

(c) Σ内の文字 c1, c2,…, cn に対し，表現 [c1c2…cn], [! c1c2…cn] は正規表現である。

(d) Σ内の文字 c1, c2, …, cn, d1, d2 に対し，表現 [c1c2…d1-d2…cn], [! c1c2…d1-d2…cn] は正規表現である。[…] あるいは [! …] の中の文字-は，d1 と d2 の文字コードの間のコードの文字 (d1 と d2 を含む) を並べて記述する省略形を表す。ただし，d1 のコードは d2 のコードより小さくなければならない。([…] の中に許されるのはΣ内の文字に限られ，[…] のネスティング等は許されない。)

(e) 文字 c あるいは，正規表現 […] に対して，c ＝あるいは […]＝は正規表現である。

(f) 文字 c あるいは，正規表現 […] に対して，c ＋あるいは […]＋は正規表現である。

(g) R および S が正規表現のとき，その合成 RS は正規表現である。RS は R と S をつなげたものを意味する。

なお，正規表現であるかぎり，(g)による合成の順番にかかわらず，先頭から

調べていくことにより，必ず(a)～(f)のどれかに該当する基本的な正規表現に分割できる。たとえば，基本的な正規表現ａ，ｂ，ｃ，ｄから((ａｂ)(ｃｄ))と合成しようと，(((ａｂ)ｃ)ｄ)と合成しようと，また，(ａ(ｂ(ｃｄ)))と合成しようと，先頭から分割できる。したがって，このような場合単にabcdと記述する。

正規表現の例：

abc　　１文字からなる正規表現ａ，ｂ，ｃを合成したもの。

ａ＊［egi-uw］？ｚ　正規表現ａ，＊，[egi-uw]，？，ｚを合成したもの。

ab＝cd［0-9］＋ef　正規表現ａ，ｂ＝，ｃ，ｄ，［０－９］＋，ｅ，ｆを合成したもの。

正規表現と文章やプログラム等に現れる文字列とのマッチングを，次のように決める。なお，正規表現中のｃはΣの文字，…はΣの文字の列を表す。

正規表現	マッチングのしかた
^	行の先頭にのみマッチする。
$	行の最後にのみマッチする。
！ｃ	文字ｃ以外の文字にマッチする。
＊	０個以上の任意の文字にマッチする。
？	１個の任意の文字にマッチする。
［…］	［　］の中のどれかの文字にマッチする。
［！…］	［　］の中の文字以外の文字にマッチする。
ｃ＝	０個以上の文字ｃにマッチする。
ｃ＋	１個以上の文字ｃにマッチする。
［…］＝	［　］の中のどれかの文字に０回以上マッチする。ただし，１回目，２回目と異なる文字にマッチしてもよ

い。(以下の3つのマッチングも同様。)

[!…]=　　[　]の中の文字以外の文字に0回以上マッチする。

[…]+　　[　]の中のどれかの文字に1回以上マッチする。

[!…]+　　[　]の中の文字以外の文字に1回以上マッチする。

正規表現と文字列のマッチングは先頭から試みられる。その各段階で正規表現＊に対応する文字列は可能な限り短くとられる。たとえば,

文字列　　caddrcddar

正規表現　　c＊r＊

のマッチングでは,正規表現の最初のcに続く＊にはaddが対応する。

これに対し,正規表現c=, c+, […]=, […]+に対応する文字列は可能な限り長くとられる。たとえば,

文字列　　caddrcddar

正規表現　　c [acdr]=r＊

のマッチングでは,[acdr]=には,addrcddaが対応する。

正規表現と文字列のマッチングの例

正規表現	文字列	マッチングの結果
c＊r	cadadr	マッチする
c＊r	cadxdr	マッチする
c？r	cadxdr	マッチしない
c [ad] r	cadr	マッチしない
c [ad]=r	cadr	マッチする
c [ad]=r	cabr	マッチしない
c [a-d]=r	cabadcbdr	マッチする

c [!ad]＝r	caaddadr	マッチしない
c [ad]＝r	caaddadr	マッチする

【参考資料】

[1] Donald E. Knuth『The TEX Book』ADDISON WESLEY, 1986

[2] Leslie Lamport『LATEX User's Guide & Reference Namual』ADDISON WESLEY, 1986

[3] 佐藤昭裕『パーソナル・コンピュータを用いたポーランド語辞書編纂の試み』古代ロシア研究，1989

[4] 島内剛一，浅本紀子『SPE 解説』『SPE 入門』岩波書店

[5] 本田道夫，中村邦彦『複数の計算機システムにおける共通コマンド体系の画面エディタの開発』香川大学経済論叢，第 60 巻第 3 号，1987

[6] 本田道夫『画面エディタマイクロ EMACS の改良』香川大学経済論叢，第 62 巻第 4 号，1990

[7] 本田道夫『Jmacs 入門』香川大学経済論叢，第 62 巻第 4 号，1990

[8] イシガキ・エム・イー・エス『bits ユーザーズ・マニュアル』

[9] ジャストシステム『一太郎リファレンスマニュアル』

[10] 高電社『Techno Mate』

[11] 日本オフィス機器『マイクロ・コスモス操作解説書』

本論中では参照番号を示してはいないが，RMACS. EXE，RLIST. EXE，RUS. SYS，FG. EXE などの各種ソフトウエアの仕様設計・開発にあたっては，MS-DOS と NEC PC-9801 に関する以下の技術資料を参照した。

[12] Igal Blumenreich『IBM PC ソフトを PC-9801 で動かすポーティング手法入門』インターフェース，№148，1989/9，CQ 出版社

[13] Igal Blumenreich『IBM PC ソフトを PC-9801 で動かすの補足：半角文字を外字登録する』インターフェース，№151，1989／11，CQ 出版社

[14] Micky『98 のフォントを取り替えませんか』The Basic 1989/12，技術評論社

[15] 浅野泰之，他『PC-9801 システム解析(上)(下)』1983，アスキー出版

[16] 阿部英志『MS-DOS プログラマーズ・バイブル』1989，CQ 出版社

[17] 川村清『PC-9801 解析マニュアル［第 0 巻］』秀和システム，1983

[18] 中島信行『MS-DOS のメモリ管理のメカニズムとその応用』インターフェース，№148，1989／8，CQ出版社

[19] 中島信行『TSR 型デバイス・ドライバの作成』インターフェース，№154，1990／2，

CQ 出版社
[20] 成田強『C言語によるデバイスドライバの作り方』1988，ラジオ技術社
[21] 藤田英詩，幸田敏記『PC-Techknow 9800』システムソフト，1984
[22] 『MS-DOS デバイスドライバ活用技法』1988，アスキー出版局
[23] 『MS-DOS とデバイス・ドライバ』トラ技コンピュータ，1989，CQ 出版社
[24] 『MS-DOS プログラマーズ・マニュアル』日本電気
[25] 『MS-DOS エンサイクロ・ペディアシステム解説編』1989，アスキー出版
[26] 『MS-DOS エンサイクロ・ペディアリファレンス編』1989，アスキー出版