パソコン用コボルエディタについて

成　久　洋　之*・井　上　弘　行**

*岡山理科大学　電子理学科

**岡山理科大学　教務部

（昭和60年９月26日　受理）

1. まえがき

昨今のパーソナルコンピュータの発展には非常に目覚ましいものがあり，４ビットCPU の開発以来，凡そ15年の歳月を経て倍々の16ビット，さらには，'80年代後半に向けて32ビット時代へと大変な進歩を遂げようとしている．本来，パーソナルコンピュータは対話型であり，しかもより手軽に使用できる簡単な言語体系としての BASIC 言語を中心に発展してきたものである．しかし，パソコンが普及すればする程，一方では大型コンピュータと同様に高級言語の利用要求が増大し，FORTRAN や PASCAL，COBOL 等の言語が使用できるまでになってきた．ところが，パソコン本来の機能を保持させる（つまり，ハンディでしかも対話型であること）ためには，エディタ機能が重要な役割を果している．そこで，単にパソコンで COBOL が利用できるというのみでなく，エディタも COBOL 的な利用体系が採られることが望ましい．現状では，パソコンで COBOL を利用するとしてもエディタは内蔵型の BASIC 中心のそれである．このことはデータの入力等の機械的作業段階での能率向上を妨げることにもなりかねない．この工程を如何に効率よく乗り越えるか否かでパソコンの利用効果，延いてはオフィス（大学事務）に及ぼす作業環境の変貌は随分と違ったものになってくる．

エディタには汎用のものもあるが，最近では，ワープロをこれの代わりに利用する動きもある．これらは，実務用向けとして数限りない機能を持っているが，複雑で一般向きに製作されているため，特定言語向きではない．ここでは，BASIC を離れ，パソコンでも走り出した実務水準の COBOL 言語の利用を主体としたエディタにつき記述するものである．

大学の事務処理の計算機に預かる比率は，今後ますます増大するであろう．そうした中で事務処理用言語，COBOL の利用環境を一つでも多く整え準備しておくことは，来るべき需要に即応できると期待するものである．

2. エディタの作動条件

本論で記述するパソコンのシステム構成は図１のとおりであり，開発マシンは日電の

PC-9801E を利用した.

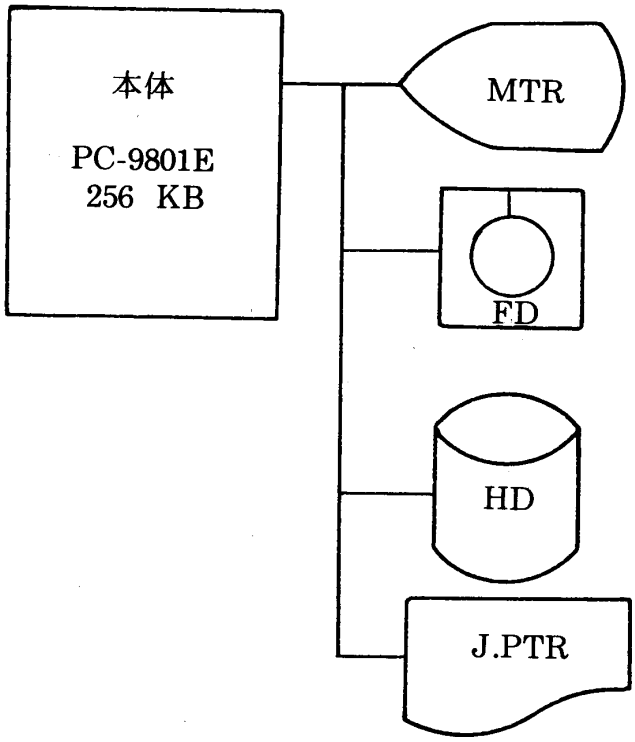


図1. パソコン装置の組合せ

標準構成としては，本体の他にモニタ，プリンタおよびフロッピーディスクである．しかし，少しでもパソコンの実用性を高め，大量の入出力が考えられるときには，ハードディスクの供え付けが最も効果的であろうと思われる．本エディタはコボルでのエディタを考慮しているので，コボルシステム類(OS 含む)は全てハードディスク上に常駐させており，常駐システムは，OS が MS-DOS (V 2.0) と必須のコボルシステム一式である．

ファイルのバックアップ及び低速で間に合う入出力については，フロッピーを使用する．プリンタについては，エディタ走行中モニタ表示と同じようにプリントアウトを配慮し，少ない機器装置群のなかで，それぞれが無駄なく有効に働くようにした。

開発システムとしては，図1のようになったが，ハードディスクは FD で代行してもよいし，プリンタも必須のものではない．メモリは 256 KB 以上が要求される．

3. 基本設計についての考え方

エディタの役目は，計算機システムと人との相互な対話ができることである．すなわち，基本的にはつぎの2つの機能が果せればよいわけである．

(1) 機械に入れる．

(2) 機械からとり出す．

エディタ設計はこの2点から出発する．開発言語をコボルのみに限定しながら利用者の立場で最も自然な形となるように次のような開発条件を考慮した．

開発条件

(1) 大型計算機に近い処理速度を引き出すこと．

(2) 会話は少ない操作で進行させられること．

(3) エディタの本質を失わない単純明快なコマンド体系とする．

(4) 画面とコマンド発行を固定付けてしまわないこと．云いかえれば，コマンド発行に自由度をもたせること．

(5) 編集は視覚的とすること．

(6) カラー機能を利用して画面を見易くすること．

(7) 操作性を重視すること．

エディタシステムの設計上，具備すべき基本的要求機能はつぎのとおり．

設計仕様

(1) テキストの編集は，いずれも内部メモリのみで行い，外部装置は使わない．

(2) テキスト入力は，コボルの正書法に則りカーソル移動キーを積極的に活用する．

(3) 基本コマンドのみで構成し単一指令，単一機能とする．

(4) ソーステキスト行とコマンド行の区別を行わないで随時，カーソル位置からコマンド発行を許すこととする．

(5) 処理の変遷（依頼，結果）は，画面を通して行われること．

まず仕様(1)については，内部メモリ上のみで行うこととし，外部装置利用による処理速度の低下を避けるためである．すなわち，コボルの特徴を有効に利用しようとするものである．

エディタ機能はテキストの編集処理が中心となるので，その内部エリア確保の仕方，あるいは用い方具合が性能を大きく左右する．その編集エリアは図２のように確保している．

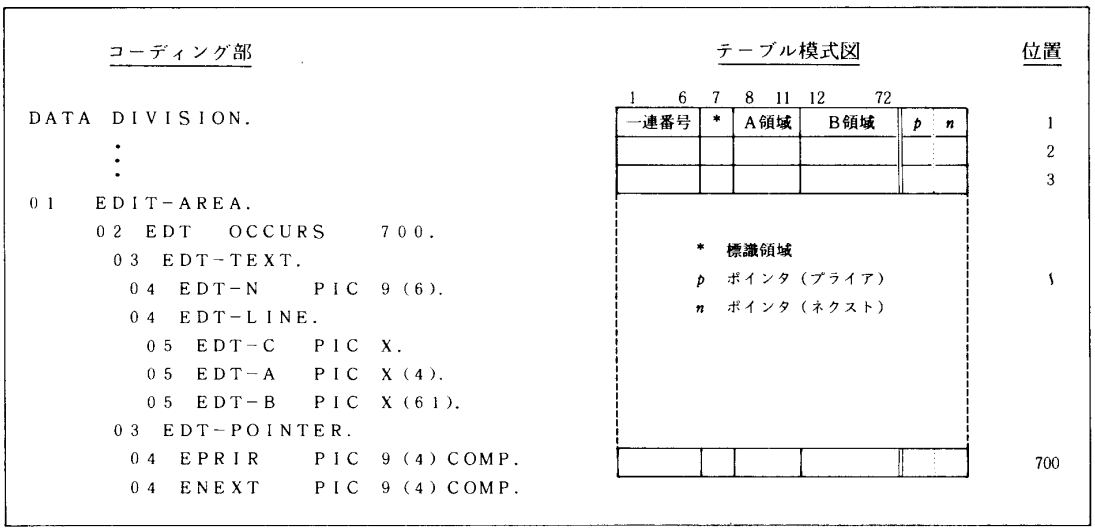


図2. 編集用エリア

ここで，位置が700で終っているのは，16ビットコンピュータのセグメント方式からの制約と，いくつかの作業用エリア確保に圧迫されるからである．テキスト順位の決定は，スライド方式を採ると頻繁なデータ移動を惹起し，速度面で損失を被るのでポインタ方式とした．この方式では，ポインタエリアが余分に要るが，ソース行の数で僅か40ステップ相当のものである．

仕様(2)については，カーソル移動の設計問題である．コボル語は，書式が一定の規則で縛られており，他の高級言語，例えばPASCALのように自由フォーマット記述は許されていない．そこで，勢いスペースキーの多用となる．これを緩和するには，カーソル移動キーを活用するのが順当であろう．幸いにして，パソコン・キーボードには，[↑]，[↓]が設置されており開発に利用したコボルでは，フィールド間の往来を許している．そこで，この問題は，フィールドの定義を工夫することで解決する．(後述参照)

仕様(3)では，必要最小限のコマンドに限定したことである．一般エディタでよくみかけるのは，細かい機能を沢山つけ加え，コマンドの豊富さを唱い文句にしているものがあるが，通常の利用においては，それほど凡てを使い切って作業を進行させることはめったにない．それよりも，本当に無ければ困る基本コマンドだけに絞って使い易く，憶え易くなるように試みた．その結果，このエディタは，11種類のコマンドに絞られた．(図9参照)これで，プログラムの編集は十分である．

最後に，仕様(4)，(5)について記述するが，これについては，バッファ構成とそのデータ定義をみることにより明らかとなる．

まずバッファについては，オペレータと計算機とが対話するときに必要な一時蓄え用のメモリで，単一入力用と複数入力用とを設けた．単一入力では，主にテキストの新規設定，追加設定などに利用し，複数入力用はテキストモディファイ等に利用するものである．このバッファ構造は図3で与えている．バッファ領域を区切っているのは，コボル記述の正書法に則るものである．すなわち，通常は，この区切りに合わせてタブ設定を行うのであるが，MS-DOS下では，この機能がない(キーとしてはあるが，これは，8桁毎の固定であってコボル向きでない)ので，前述したようにカーソル移動キーと，この定義により，コボル正書法用のタブ機能に代わるものとして模擬的に実現させた．

例えば，7，12，30，72にタブ設定を考えると，データ名を，一例として，

```
01  BUFFER.
    02  FILLER      PIC  X(6).
    02  BUF7        PIC  X(5).
    02  BUF12       PIC  X(18).
    02  BUF30       PIC  X(42).
    02  BUF72       PIC  X.
```

と定義しておき，

ACCEPT BUFFER AT LOW-COL.

なる命令を発行するだけで可能となるものである．

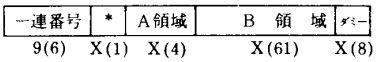
LOW-COL は，画面上の開始点であり p とすると，

定　義

```
single
 01 SCREEN-BUFFER.
    02 SBF.
     03 SBF-TEXT.
      04 SBF-N        PIC 9(6).
      04 SBF-LINE.
       05 SBF-C       PIC X.
       05 SBF-A       PIC X(4).
       05 SBF-B       PIC X(61).
     03 FILLER        PIC X(8).


plural
 01 SCREEN-TBL.
    02 TBL  OCCURS  24  INDEXED  ST.
     03 TBL-TEXT.
      04 TBL-N       PIC 9(6).
      04 TBL-X REDEFINES TBL-N
                     PIC X(6).
      04 TBL-LINE.
       05 TBL-C      PIC X.
       05 TBL-AB.
        06 TBL-A     PIC X(4).
        06 TBL-B     PIC X(61).
     03 FILLER       PIC X(8).
```

構　造



* 標識領域

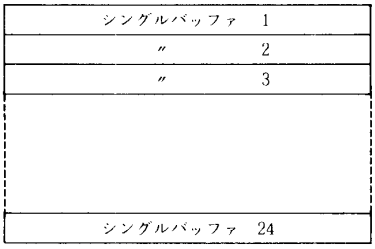


図3．入出力バッファ

$p=100*l+c$（l：行，c：桁）（$1\leqq l\leqq 24$，$1\leqq c\leqq 80$）

で与えられ

$$101\leqq p\leqq 2480 \quad \text{且つ} \quad p-\left[\frac{p}{100}\right]*100=1$$

をみたす値であればよい．

レコード名の先頭は，無名データ項目としているが，これにより無用なフイールドは自動スキップとなる．

こうした機能は，画面上で行う対話に対し勝れたプログラミング環境を確立しておりプログラム作成上，高く評価に値するものである．

図3で SCREEN-TBL（複数バッファ名）に入ってくるものは，ソーステキスト行であってもよいし，それがコマンド形をしておってもよい訳で，こうして，この2つの区別は（見かけ上）行われていないかのように作動する．ただ機能上は，その区別を無視することはできないので，そのためには手続部に判定文を挿入して実処理に供えている．

4. エディタプログラムの構造

本エディタを構成しているプログラムは，メインプログラムの他に３つのサブルーチンを有しており図４のとおりである．

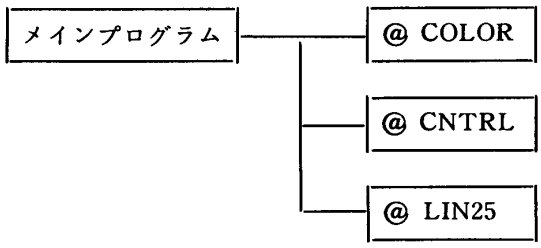


図4. プログラムの構成

メインプログラムでは，エディタの主務であるコマンドの受け付けと，それらを解析・遂行して応答することである．11個用意したこれらの要求コマンドと処理セクションは，図５のように構成されている．これらを機能的に結び，又支援するため図６のような各処理を実施する．

MAIN セクションでは，初期画面の対処，コマンドの受付・解析，モディファイ処理

コマンド	セクション名	処　　理
I	INSERT−SUB	テキストの設定・追加
T	TYPE−SUB	一画面分表示
S	SEARCH−SUB	文字列探索
D	DELETE−SUB	行消去
F	FB−SUB	前方表示
B	FB−SUB	後方表示
N	NUMBERRING−SUB	キーナンバ付け
M	MOVE−SUB	移動（消去）
C	COPY−SUB	移動（保存）
L	LIST−SUB	リストアウト
E	END−SUB	終了

図5. コマンド節

節　　名	処　　理
MAIN	(i) 初期画面処理 (ii) コマンドの受付 (iii) モディファイ処理 (iv) セクションの制御
FNSET−SUB	ファイル名セット
MODE−SUB	モードライン（25）の表示
ANALYSIS−SUB	コマンド解析
POTION−SUB	カレントライン，ドライブラインの割り出し
COMN−SUB	増分値割り出しと表示

図6．主節と副節

及び各コマンドへの制御を受け持っている．この中で，コマンドの解析には64ステップを要したが処理速度では気になる程の遅延もなく，応答は殆んど反射的である．

モディファイ処理では，いわゆる，編集中のプログラムを修正するもので，これはコマンド方式としておらず，画面バッファに入ってくるものを直接にメモリ上の編集テーブルに引き渡し，通常云われるところの〝差し替え〟で新旧の交換を行っている．

コマンド受け付けとモデイファイ機能実現とについては，表面上区別していないので画面の任意行から投入が許されると同時に，修正も任意行の任意位置をもって実行可能である．このことは，このエディタの大きな特徴で方法的には，画面そのものをレコードとし直接にデータとして定義する．そして，ステートメント自身が，そのうえを走査できる仕組みによっている．このコーデイング部を図7に示すが，これは他言語にみられない手法で，多方面に応用できるものと思われる．

図7では表示テキストを SCREEN-TBL に納め，外部操作入力は，TELPORT-TBL に受ける．命令部はこれだけのものであるが，TELPORT-TBL に ACCEPT することで，その定義記述から任意位置での入力が可能となるのである．さらに，手続部の IF 文で実際に入ってきたのがコマンド入力であったかどうかを調べている．そして，引き続く命令では，編集テーブル上に移動を起こしており，LOW-COL には，オペレータの最終操作位置が自動格納されることになっている。

3つのサブルーチン

@ COLOR，@ CNTRL，@ LIN25

については，それぞれカラー指定，画面特殊制御，行25の取り込みからなっている．サブ

```
データ部
 01  SCREEN-TBL. ……画面バッファ
     02  TBL  OCCURS  24  INDEXED  ST.
       03  TBL-TEXT.
         04  TBL-N        PIC  9(6).
         04  TBL-X  REDEFINES  TBL-N
                          PIC  X(6).
         04  TBL-LINE.
           05  TBL-C      PIC  X.
           05  TBL-AB.
             06  TBL-A    PIC  X(4).
             06  TBL-B    PIC  X(61).
       03  FILLER         PIC  X(8).
 01  TELPORT-TBL  REDEFINES  SCREEN-TBL.……
     02  PRT  OCCURS  24.                          対話窓口
       03  FILLER         PIC  X(6).
       03  PRT-LINE.
         04  PRT-CAB      PIC  X(66).
       03  FILLER         PIC  X(8).
手 続 部
       :
       :
       ACCEPT  TELPORT-TBL  AT  LOW-COL.
       IF      TBL-C(LOW) = "x" GO  TO [コマンド処理先] .
       MOVE    TBL-TEXT(LOW)  TO  [編集テーブル] .
       :
       :
```

図7. コマンド受付とモディファイ機能

ルーチンといえば，第一に実行速度が議論されるもので，本来ならばアセンブラで記述されるところであろうが，既述したようにコボルのパソコン能力への接近を試みているため，敢えて言語を特定した．その実行速度の問題点であるが，@ COLOR に関しては，とり上げるものはない。問題が生じたのは，@ CNTRL と @ LIN25 である．

@ CNTRL でみると，カーソルのポジショニングとかオンベルなどでは，何の障害も感じないが，特定範囲の画面クリア或いはスクロールを行う場合には，それ相当のメモリ操作が伴う訳で，動作状態に目がついて行く速さである．この点は，一考を要する．

また，@ LIN25 では，この行へディスプレイすると画面が自動スクロールする．これでは，スクリーンエディタとしての使命が喪失してしまう．それを回避するため少々命令が重なり，処理速度に難が生じた．制御をもっとよく検討すれば，良策があるのではないかと思う．

最後に，これまでに出てきたバッファ間，編集テーブル間等の関係をソーステキストの

流れのうえから簡単にまとめると図8のようになる．対話窓口というのは，各コマンド指令が入ってくるところで，コボル言語では，画面バッファに重畳させて定義している．こ

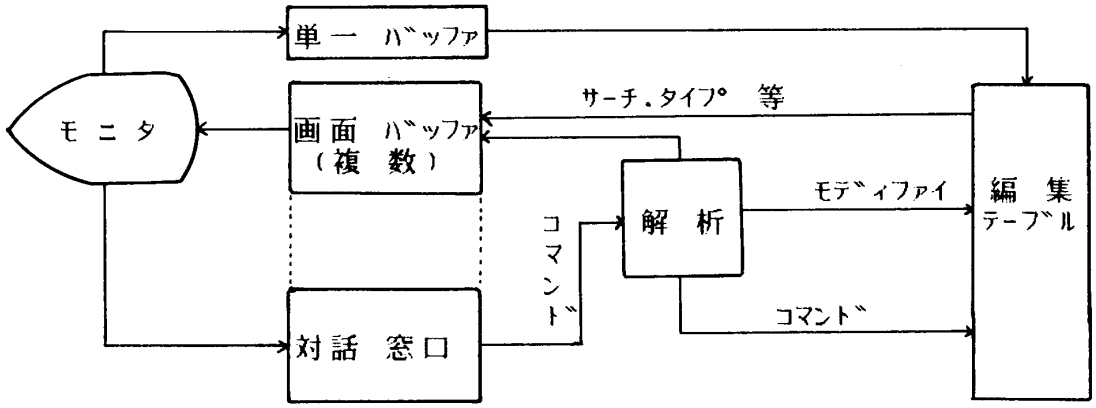


図8. 指令とテキスト

こから，凡ての機能要求が受け入れられる．要求がコマンドであるか，モディファイであるか等を解析し，それに応じた編集テーブル上への処置を施す．ある種のコマンドによっては，画面バッファにため込み，一挙にモニタ表示するようにしている．サーチコマンドに関してのみは，処理上の時間を配慮して，目的のものが発見され次第，その都度モニタ表示（画面バッファ）するようにしている．テキスト入力は，単一バッファから直接編集テーブルに入って行くが，複数バッファにも送られて後続のコマンドに供えるようになっている．

5. エディタの起動とコマンド

基底ドライブをAとした場合，ここにシステム，エディタ，取り扱いファイル等の凡てを集中させ，

A>RUN EDIT name ↵

として起動する．nameは，編集しようとするソースプログラムのディスク上のファイル名である．MS-DOS 下では，name.CBL の形をしているが，拡張子の指定は省略可(指定不能）としてある．

編集ソースプログラムファイルは新旧不問である．バックアップは，EDLIN のようにはなっていないので，必要ならば，ユーザの責任下で管理するものとする．

コマンドは，いずれも

$c[l_1[,l_2]]$

c：指令コード（I, T, S, D, F, B, N, M, C, L, E）

l_1：作用開始キーナンバ（6桁以内）

l_2：作用終了キーナンバ（6桁以内）

（注：キーナンバとは一連番号のことであり，空白の使用は自由である.）

の単一書式が与えられているのみである．憶え易さと簡便さを配慮した．ここで，コマンド c↵とした場合，キーナンバはコマンドラインの一連番号が l_1 を担う．

コマンドの種類と機能は，図9のとおりである．

呼び名	形　式	動作処理・機能
インサート	I; Il_1	ソース行を編集ファイルに追加・挿入
タイプ	T; Tl_1	ソース行表示（24行分）
サーチ	Sl_1; (Sl_1,l_2)	l_1 (l_1, l_2) を文字列として探索表示する
デリート	D; Dl_1; Dl_1, l_2	行削除
前方表示	Fl_1; ↵	指定行から一画面分の表示，継続繰り返しは↵
後方表示	Bl_1; ↵	指定行から一画面区切りで後方（先頭）表示
ナンバリング	N; Nl_1	キーナンバ（一連番号）の振り直し（空白時は必須）
移動（消去）	Ml_1; Ml_1, l_2	指定ソース行をコマンドラインの直前にもってくる
複写（保存）	Cl_1; Cl_1, l_2	指定ソース行と同一のものをコマンドラインの直前に生成
リスト	Ll_1, l_2	プリンタアウト
エンド	E	終了
モディファイ		任意修正

（注：コマンドの作用は，ＥＤＬＩＮに合わせ，指定するラインの直前に向けて行われる）

図9．コマンドの種類と許容形

前述したが，一般エディタにみられるようにコマンドの種類は豊富なものではなく，むしろ簡潔にまとめ，誰でもが全容を摑めるようにした．従って，マクロにみて，外部ファイルのマージングや編集中におけるソースファイルの消去はとり上げてない．また，ミクロなとこでは，文字列置換とか作用範囲の限定，或いはキーナンバの任意値付与，行内文字列移動といったようなものも一切割愛した．さらに，CTRL キーとか ESC キーとかによる複合キーコマンド形式はとらなかった．ただ，日本語入力に関しては，OS レベルの制約で CTRL + XFER のキー操作を強要されるが，これは致し方のないものである．

6. 結果とその検討

本論で記述したエディタでソースプログラム長368ステップのものについて時間特性をみるとつぎのような結果が得られた．

(1) ディスク上コピー　　4秒（同一ドライブ）
(2) エディタ　ロード　　9秒（約100 KB）
(3) 編集プログラム　メモリイン　　5秒
(4) ナンバリングコマンド　2秒（全ステップ並びに一画面分表示）
(5) タイプコマンド　　3秒（24行分表示）
(6) エンドコマンド　　5秒（HD アウト）

なお，このソースのコンパイル時間は56秒であった．すなわち，ほとんどのものが数秒～10秒内位で応答を完了し，デリートコマンドのようなものにあってはカウントにすら上ってこない．このことは，作業進行上まずまず問題とする程の事でなく，大型コンピュータと同等の感覚で利用できることを考えれば，一応満足のできるものである．

また，一連番号を持たないコボルソースでも，本エディタは受容可としている．このとき，仮りに他システムからのものを移植したいような場合でもその儘容易に対処できる．これはソフト流用上，見逃がすことのできない機能と思う．

コマンドの利用頻度では，タイプ関係は以外と多い．測定では3秒となっているが，一層の検討を加え，1秒位迄には持って行けないものかと考えている．これを凌駕すれば，このエディタも相当なものとして受け止めてよいと思う．

このエディタはコボルのみで開発されているという最も大きな特徴を持たせた．その結果，言語間互換は高く，大型コンピュターとかオフィスコンピュターのレベルのシステムへも，比較的容易に移し替えられるものと思われる．また，他機種のパソコンに対しては，8086系はもとより当システムがサポートされるところでは，ソースレベルにおいて容易に利用できるものと思われる．なお，今後の研究方向としては，

○バックアップ・ファイルの確保
○外部ファイルの取り込み（マージ，切り換え）
○大規模ソースに対する外部装置の起用
○画面の高速処理
○特殊処理の改良
○ガーベッジ問題

等が考えられる．

このように，未解決部分も残してはいるが，ベーシック一辺倒で，或いは，ワープロ機能の波に乗って普及してきたパソコンを，それだけで終らせることなく，より進んで多角的に活用したいものである．OA 化の波は，必ずしも企業の占有物ではあるまい．必ずや大学事務にも活用の場は，今後ますます増大するものと思われる．そうした中において，このエディタもしくはコボルが日常的となり誰でもに使えるパソコンとなり，これからの大学の OA 化にいささかでもふさわしくあればと願う次第である．

参考文献

〔1〕 日本電気「日本語 LEVEL IICOBOL マニュアル」
〔2〕 篠原幸悟「日本語 LEVEL IICOBOL 入門」パソコンワールド No. 5, pp. 58～61 (1984)
〔3〕 日本電気「MS-DOS ユーザーズマニュアル」pp. 195～200.
〔4〕 南澤宣郎「OA 革命」日本経済新聞社
〔5〕 長野時男「漢字 L IICOBOL 入門」電波新聞社
〔6〕 井上弘行「スクリーンエディタ (1)」information Vol. 4, No. 7, pp. 115～131 (1985)
〔7〕 井上弘行「スクリーンエディタ (2)」information Vol. 4, No. 8, pp. 74～78 (1985)

A Cobol Editor for the Personal Computer

Hiroyuki NARIHISA*, Hiroyuki INOUE**

**Depertment of Electronic Science*
***Division of Academic Affair*
Ridaicho 1-1 Okayama 700, Japan

(Recieved September 26, 1985)

ABSTRACT

In this paper, we present COBOL Editor for Personal Computer. In the business processing area, many kinds of personal computers are developped. However, thir editors are mainly used by only BASIC language. Practical points of view, we often use COBOL language for business processing task on the middle or large scale computer system. Accordingly, if we are able to use COBOL editor for Personal Computer, the efficiency of I/O operation throug the personal computer as a terminal unit to host computer would be increased.

Here, we developped COBOL Editor for personal computer by using "JAPANESE KANJI LEVEL II COBOL". This editor is composed of 11 kinds of commands and its usage is very easy. The command format is following.

c[11[, 12]]

here c : commands (are I, T, S, D, F, B, N, M, C, L, and E.)
11 : start key number
12 : end key number
(notice : key number is sequential number on the source program from 1 to 6 column.)

Moreover, we can exchange the given source cobol text to the new-line text only by using key-board (return key) without using the above mentioned commands.