



プログラミング教育用擬似計算機システムについて*

的 場 裕 司** 吉 岡 信 夫** 佐 藤 武 久**

Abstract

It is very difficult to show how to teach fundamental concepts about computer programming to introductory programming courses. In this paper, we propose the Pseudo Machine System (PCOM) for programming education which is simulated by PDP-11/20.

It is available interactively and simultaneously for some students, and has a very simple instructionset and an assembly language (ESPL). We describe here its facilities, good points and how to use it.

1. ま え が き

米国においては、プログラミング教育にはいわゆる高級言語がよいとされ¹⁾、各大学からのアンケート調査でも、プログラミング教育における学生の使用言語は FORTRAN が約 70% を占め、以下 BASIC が 13%、PL/I が 8% で APL, ALGOL, COBOL およびアセンブリ言語はそれぞれ 3% 以下という報告がされている^{2),3)}。しかし、一方では機械語またはアセンブリ言語によるプログラミング教育についても種々の教育用システムが稼動しているようである^{4),9)}。これらをもととみると、実存する計算機の命令をそのまま利用するものはすくなく、教育に適した命令を持った擬似計算機 (Pseudo Machine) を想定し、それを実存の計算機で処理する方法が多く用いられている。そのため、擬似計算機の内部表現も 10 進数を用いたものが多く、また、擬似計算機を実現するために利用する計算機としては大型計算機が多いためかバッチ処理されているものが多い。

一方、我が国のプログラミング教育の現状をみると、東京大学、京都大学等に設置されている大型計算機センタは教育目的には直接利用できないことになっており、実際に教育目的に利用されている計算機は中型機以下のものがほとんどであり、しかも、使用言語は

FORTRAN が中心のようである。しかし、機械語またはアセンブリ言語を教育するためのシステムもいくつか作成されており、これらにも擬似計算機概念が採用されている¹¹⁾⁻¹⁵⁾。

さて、今後のプログラミング教育を考えると高級言語による教育はますます重要なものとなるのはもちろんであるが、ミニコンピュータやマイクロコンピュータがめざましい発達をとげ、それらを利用する機会も多くなりつつある現在、機械語またはアセンブリ言語による教育の重要性もより以上に増大しつつあると考えられる。そこで、比較的コストの安いミニコンピュータを利用し、複数のユーザが会話形式で機械語またはアセンブリ言語によるプログラミングの教育を受けられ、計算機の基本概念であるプログラム記憶方式の基礎を理解させるための教育用システムを試作したので以下に示す。

2. 教育目的と教育方法

機械語またはアセンブリ言語の教育用システムとしては、実存する計算機を直接利用するよりも擬似計算機を利用するほうが種々の点で適しており、とくに擬似計算機としては Knuth の MIX 計算機¹⁰⁾が一般性があってよいという意見がある⁹⁾。しかし、何をどの程度教育するのがよいかということになると、教育者によりすこしずつ考え方にも方法にも相違があるのが当然といえよう。

計算機のプログラミング教育に擬似計算機を利用す

* Pseudo Machine System for Programming Education by Yaji MATOBA, Nobuo YOSHIOKA and Takehisa SATO (Faculty of Engineering Science, Osaka University).

** 大阪大学基礎工学部情報工学科

る場合にも、その機能をどの程度にするかは大いに意見のわかれるところである。もちろん、擬似計算機をシミュレートするための計算機の機能による影響がもっとも大きいかもしれないが、擬似計算機の機能を大きくするとそれだけ複雑な処理が可能となるが、逆に初心者にはかえって負担が多くなり教育効果をさまたげる原因ともなりかねない。したがって、擬似計算機による教育を考える場合も初心者用、中級者用、高級者用というように数種類の擬似計算機を利用するのが理想的であろう。ここでは、初心者用に限定し、しかも、理解しやすいことおよび使いやすいことに重点をおいて考えることにする。

われわれのシステムの目的とするところは、プログラミングの初心者に計算機内部の各種レジスタの内容の変化やその機能、記憶装置の番地とその内容について、および、命令の実行順序等の理解を容易にさせることにある。したがって、ここで考える擬似計算機は

- (i) 1語で1命令を構成する1アドレス方式
- (ii) その内部表現は10進数
- (iii) 簡単な命令セットの採用
- (iv) 数値は整数のみ

というかなりきびしい制限がついている。

また、利用者の立場を考えて上記の目的をより効率よく達成するために、

- (i) 同時に複数個の端末より利用できる
- (ii) テレタイプにより会話形式で利用できる
- (iii) 簡単なアセンブリ言語が利用できる
- (iv) アセンブリ言語による命令は1命令入力するごとにチェックされる
- (v) いつでもプログラムの修正、変更が可能である
- (vi) プログラムの実行に際しては1命令ずつ実行させることができ、各種レジスタや特定の番地の内容の変化が確認できる

というような機能が付加されており教育に便利なように考慮されている。

3. システムの機能

目的をプログラミングの初心者の基礎教育にしばったため、使用する擬似計算機（以下PCOMと呼ぶ）もごく簡単なものである。PCOMはFig. 1に示すように同時に3端末から会話形式で利用可能なように作られており、各ユーザはそれぞれ1,000語のメモリが利用できるようになっている。また、PCOMのプ

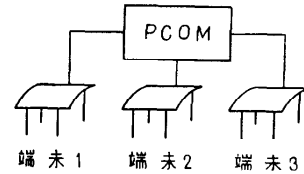
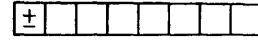


Fig. 1 PCOM System.

数値の表現



$-(10^7-1) \sim 10^7-1$ の整数

符号：正 (0), 負 (1)

命令の表現



(i) (ii) (iii) (iv)

(i): 間接番地指定 (1または0)

(ii): インデックスレジスタ指定 (1または0)

(iii): 命令部分

(iv): 番地部分 : 未使用 (0)

Fig. 2 Number and Instruction of PCOM

ログラミングは基本的な命令のみで構成されたアセンブリ言語 ESPL (Educational Symbolic Programming Language) および 10 進数表示された機械語が用いられる。

PCOM の 1 語は 10 進数 8 桁で構成され、数値および命令は Fig. 2 のようになっている。命令部分には Table 1 (次頁参照) に示した PCOM の命令のうちどれかが対応することになり、記憶容量は 1,000 語である。レジスタとしては演算用レジスタ (Acc), インデックスレジスタ (X_R), 命令レジスタ (I_R) およびインストラクション・カウンタ (I_C) が各 1 個ずつある。このうち Acc と I_R は 10 進 8 桁 (含符号), X_R と I_C は 10 進 3 桁 (符号なし) の容量がある。

ESPL を利用する場合には Table 1 に示した命令のほか Table 2 (次頁参照) に示す擬似命令が利用できる。ESPL でのプログラムの表現はフリーフォーマットで端末のテレタイプから入力できる (付録② (P. 123 参照))。そのとき、1 つのステートメントは 72 文字以内で表現し、その終りは "RETURN" で示すステートメントはラベル、オペレータおよびオペランドと呼ばれる部分から構成され、ラベルとオペレータはコロン (:) によって区切られ、オペレータとオペランドは 1 つ以上のブランクによって区切られる。ラベ

Table 1 Instruction Sets of PCOM

	命令		モード	機能
	記号	コード		
転送命令	LDA	10	1	$(m) \rightarrow Acc$
	STA	11	1	$(Acc) \rightarrow m$
	LDX	80	2	$(m) \rightarrow X_R$
	STX	81	2	$(X_R) \rightarrow m$
演算命令	ADD	12	1	$(Acc) + (m) \rightarrow Acc$
	SUB	13	1	$(Acc) - (m) \rightarrow Acc$
	MLT	14	1	$(Acc) \times (m) \rightarrow Acc$
	DIV	15	1	$(Acc) \div (m) \rightarrow Acc$
	ADX	82	2	$(X_R) + (m) \rightarrow X_R$
	SBX	83	2	$(X_R) - (m) \rightarrow X_R$
分岐命令	BRN	40	1	m へブランチ ($m \rightarrow Ic$)
	BMI	41	1	$(Acc) < 0$ なら m へブランチ
	BPL	42	1	$(Acc) \geq 0$ なら m へブランチ
	BEQ	43	1	$(Acc) = 0$ なら m へブランチ
	BNE	44	1	$(Acc) \neq 0$ なら m へブランチ
	JSR	45	1	$(Ic) \rightarrow X_R, m \rightarrow Ic$
	RTS	46	1	$(X_R) \rightarrow Ic$
	BNX	85	2	$(X_R) \neq 0$ なら m へブランチ
入出力他	GET	01	1	データを m へ入力
	PUT	02	1	(m) を出力
	HLT	00	0	ストップ

(dd): dd の内容を意味する

m: 実際の番地を意味する

モード1: 間接番地, インデックスレジスタ共に指定可能

2: 間接番地のみ指定可能

0: 番地部は関係なし

Table 2 Pseudo Operation Code of ESPL

記号	機能
ORG	以下の命令をセットする番地を指定する
DEC	10 進数をセットする
WKS	作業用番地を指定した個数だけ確保する
ADR	アドレス・コンスタントを定義する
END	ESPL によるプログラムの終りを示す

ルはアルファベット2文字以内の組合せで示され、オペレータは Table 1, 2 に示した記号表現が用いられる。オペランドは "@{シンボル, 絶対番地または数式表現}, X" の形で表わされる。@とXはそれぞれ間接番地とインデックスレジスタ修飾を指定した場合にのみ使用される(付録⑤)。シンボルはラベルとして定義されたものと命令が入っている番地そのものを示すための記号 "*" があり(付録⑥)、絶対番地は0~999の整数で示される。数式表現とはシンボルと絶対番地を+または-で結合したものをいう(付録⑥)。

PCOM を実際にシミュレートしているのは PDP-11/20 計算機である。PCOM のために実際に利用しているハードウェアは PDP-11/20 CPU, 24k 語(16ビット/語)のメモリ, 1.2M 語のディスクおよびコンソ

Table 3 Command List of PCOM

コマンド名	機能	例
BLOG	PCOM の Initial Set	BLOG
ELOG	DOS にコントロールを返す	ELOG
HEllO	CMD モードにする	HE
EDit	プログラムの入力, 編集を行う	ED
LIst	入力プログラムのリストを出す	LI
ASsemble	アセンブラを働かせる	AS
RU n xxx	xxx番地より命令を実行する	RU 100
TEst xxx	xxx番地よりテスト・ランを行う	TE 200
BYe	READY モードにする	BY

ールを含めた3台のテレタイプである。これらのハードウェアにより PDP-11/20 はディスク・オペレーティング・システム(DOS)の管理のもとで稼動しており、したがって、PCOM もこの DOS の管理下で働く1つのプログラムということになる。

PCOM は会話形式で利用できるようになっているのでユーザはそのためのコマンドを使用しなければならない。端末としてはテレタイプが使用されているので、これが PCOM からの出力およびユーザからの入力のために使われる。コマンドの一覧表を Table 3 に示す。これらのコマンドは使用可能な状態が異なるものがあり、その関係を示したものが Fig. 3 である。

BLOG および ELOG コマンド(付録①, ⑩)は PDP-11/20 のコンソール用テレタイプ(端末1)からのみ入力可能で、DOS モードから PCOM 使用可能な状態(READY モード)にするためとその逆の動作を行なうために使用される特殊コマンドである。一般のユーザは READY モードで HE コマンド(付録②)を入力することにより PCOM との会話が可能となり(CMD モード)、BY コマンド(付録⑬)を入力することにより PCOM との会話の打ち切りを知らせる。

CMD モードにおいて、LI(付録⑧)、RU および AS コマンド(付録⑨)が入力されると、それに対応する処置がとられ、処置が終了するとやはり同じモードにもどる。これにたいして、TE コマンド(付録⑭)が入力されると TEST モードに変化する。ここでは、

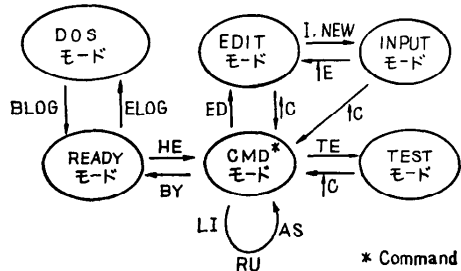


Fig. 3 Command State Diagram of PCOM

1 ステップずつ命令を実行させ、各種レジスタの内容をチェックするためのサブコマンド P (付録③) と特定の番地の内容をチェックするためのサブコマンド D (付録④) が使用される。ED コマンド (付録③) が入力されると EDIT モードに変化する。ここではプログラム編集用のサブコマンドが利用できる。これらのうち、NEW (付録④) および I サブコマンドにより INPUT モードに変化し、新しいプログラムおよび修正用プログラムの入力が可能となる。このモードで ↑E (コントロール E) が入力されると EDIT モードにもどる。また、いかなるモードにおいても ↑C (コントロール C) が入力されると CMD モードになる (付録⑦)。

4. システムの構成と内部処理

PCOM をシミュレートしているハードウェアについては前述したが、PCOM そのものは PDP-11/20 用のマクロアセンブリ言語 (MACRO-11) で書かれたプログラムで、その構成は Fig. 4 に示すように各種のモジュールからなっている。PCOM を利用するときはこれらすべてのモジュールはメモリ内に常駐することになる。DOS が使用する領域が約 4k 語、各端末から共通に利用されるプログラム領域と、データ領域のうち共通データ領域とスタック領域との合計は約 7k 語であり、各端末専用のデータ領域にはそれぞれ約 3k 語が用いられている。したがって、現在の PCOM では各端末から入力可能な ESPL によるステートメントは最大 100 程度の大きさに限られる。

PCOM を多端末から同時に使用可能とするために各モジュールのうち、INITIAL SET ルーチンと入出力処理ルーチン以外はすべて再入可能型に作成されており、入出力処理ルーチンは逐次再使用可能型となっている。

DOS システムルーチン
INITIAL SET ルーチン
端末切換えルーチン
入出力処理ルーチン
エディタ
アセンブラ
インストラクション・インタープリタ
メイン・ルーチン
共通データ領域
端末1用データ領域
端末2用データ領域
端末3用データ領域
スタック領域

Fig. 4 PCOM Modules in memory

INITIAL SET ルーチン

PDP-11/20 において DOS から PCOM へ切りかえることにより、DOS が使用しているメモリの一部を修正して利用する必要があるため、そのようなエリアや各種レジスタ類をスタック領域に退避させるルーチンである。PCOM の使用が終了すると ELOG コマンドによりこれらの退避されていた情報が DOS が必要とするエリアにもどされることになる。

メイン・ルーチン

BLOG コマンドにより DOS モードから PCOM の READY モードに切りかえ、各端末から PCOM を使用可能な状態にセットする仕事および種々のコマンドをチェックする仕事から構成されている。

端末切換えルーチン

このルーチンが起動されるのは PDP-11/20 の CPU のコントロールの受け渡しが行われるつぎのいずれかの場合である。

(i) 端末が入出力待ちの状態となった場合

入出力待ち状態となった端末に対応するフラッグを 1 にセットしてから、フラッグが 0 の状態にある他の端末をさがし、もしみつければその端末にコントロールをわたす。すべての端末が待ち状態のときは、どれかの端末のフラッグが 0 になるまでループして待つ。

(ii) インストラクション・インタープリタにおいてユーザの作成した PCOM のプログラムの 1 命令を実行し終った場合

この場合は他の端末がすべて入出力待ち状態 (フラッグが 1) ならばコントロールの移動はない。しかし、どれかが入出力待ち状態でない (フラッグが 0) とときはその端末にコントロールがうつる。

(iii) 端末から入出力割込みがかかった場合

各端末からの入出力割込みは PDP-11/20 のハードウェアによって処理され、端末固有の番地にブランチし、そこで各端末にたいする必要な処理をして (入出力処理ルーチン) コントロールは割込み発生時の状態にもどる。したがって、割込みを発生した端末にその時点でコントロールがうつるとはかぎらない。

入出力処理ルーチン

端末から入出力割込みが発生するとこのルーチンへコントロールがうつる。このルーチンは最高の優先度で処理されるようになっていて、このルーチンを実行中はすべての割込みは禁止状態にある。このルーチンでは各端末から入力される 1 文字あるいは各端末へ出力される 1 文字についての処理を行うだけであ

る。したがって、このルーチンを実行中は他の割込みを禁止してもほかへの影響はほとんどないものと考えられる。

エディタ

ED コマンドにより起動されるルーチンで、各端末用のデータ領域にソースプログラム（ステートメントの集合）を読み込むことと読み込まれたソースプログラムを修正することがおこな仕事である。ある端末から一度入力されたソースプログラムはEDITモードにおけるサブコマンドであるNEWが入力されるまでその端末に対応するデータ領域に存在する。

このルーチンで処理されるサブコマンドは大別すると、データ領域にあるソースプログラムの各行を指定するためのポインタを制御するもの、ソースプログラムの修正を行うもの、および、ポインタの示す行を出力するものにわけられる。それめの一覧表を Table 4 に示す。

アセンブラ

AS コマンドにより起動され、端末個々のデータ領域にあるソースプログラムをアSEMBルし、PCOMの機械語に変換し、PCOMのメモリへロードする仕事を行う。各ステートメントの表現方法のミスはINPUTモードにおける入力時にチェックされているので、ここではラベルの多重定義のチェック、シンボル・テーブルの作成、オペランド部の値の計算、機械語の作成および指定された番地への機械語命令のセットが中心となる。アSEMBリ・リストやシンボル・テーブル・リストを出力するかどうかはアSEMBラからの質問にたいしてYESかNOで答えることができるようになっている（付録⑩、⑪）。

インストラクション・インタープリタ

RU または TE コマンドにより起動され、PCOMの指定された番地の内容を解釈し、PCOMの命令として実行するルーチンである。1つの命令の解釈、実行が終ると端末切換えルーチンへコントロールはうつる。

Table 4 Subcommand List of PCOM Editor

ポインタ制御用

B: ポインタをソース・プログラムの最初の行にセットする

E: ポインタをソース・プログラムの最後の行にセットする

A±n: ポインタを現在の位置から n 行進める (もどす)

ソース・プログラム修正用

Kn: ポインタが示している行を含めて以下 n 行を消去する

I: ポインタを示している行の前に修正用プログラムを挿入する

NEW: 現在のソース・プログラムをすべて消去し新しいものを入力する

チェック用

V: ポインタが示している行を出力する

5. む す び

ここでは、ごく簡単な擬似計算機(PCOM)を利用した初心者むけのプログラム教育に関する一方法について述べたが、これにより計算機の基本概念を理解させるには一応十分ではないかと考えられる。したがって、リロケータブルな概念、マクロ、リンケージ・ローダ、割込み機能や記憶保護機能等についての教育はここで述べたような擬似計算機による基本教育のうち、実存する計算機を利用することで解決してもほとんど不都合はないと思われる。

なお、現在のPCOMシステムについていえば、端末がテレタイプであるためエディタ用サブコマンドがすこし複雑であることと命令の種類がすくないことが難点といえよう。また、付録に示したようにTEコマンドによる出力において(付録⑩)、命令部に関してはその記号表現を入れ、インデックス・レジスタや間接番地を指定された命令については実効アドレスもあわせて示すのがよいと考えられる。このほかユーザに関する各種のデータがとれるようにシステムを改善することにより、より本格的な教育用擬似計算機システムを構成することが可能となろう。しかし、現在のところ使用経験が浅く、どのようなデータをどのような形式で集めればよいかまで検討していないので今後の研究にゆずることとする。

最後に、日頃より御指導いただいている本学藤沢俊男教授に感謝致します。また、本研究の一部は文部省科学研究費によるものである。

参 考 文 献

- 1) D. Gries: What Should We Teach in an Introductory Programming Course?, SIGCSE, Vol. 6, No. 1, pp. 81~89 (1974).
- 2) A. Ralston: The Future of Higher Level Languages (in Teaching), International Computing Symposium, pp. 1~10 (1973).
- 3) Andries van Dam et al.: A Survey of Introductory and Advanced Programming Courses, SIGCSE, Vol. 6, No. 1, pp. 174~183 (1974).
- 4) S. Reisman: A Survey of Pedagogical Programming Languages, SIGCSE, Vol. 5, No. 2, pp. 13~21 (1973).
- 5) T. D. Sterling & S. V. Pollack: Teaching Simulators or Ideal Teaching Machines, SIGCSE, Vol. 6, No. 2, pp. 45~56 (1974).
- 6) S. C. Shapiro & D. P. Witmer: Interactive

Visual Simulators for Beginning Programming Students, SIGCSE, Vol. 6, No. 1, pp. 11~14 (1974).

- 7) J.R. Mashey et al.: A Self Modifiable Assembler for Instructional Purposes, Proceedings of the ACM 25th Anniversary Conference, pp. 310~312 (1972).
- 8) J.R. Mashey: ASSIST: Three Year's Experience With a Student-Oriented Assembler, SIGCSE, Vol. 5, No. 1, pp. 157~165(1973).
- 9) E.M. Greenawalt & D.I. Good: The MIX Computer as an Educational Tool, Proceedings of the ACM 25th Anniversary Conference, pp. 302~309 (1972).
- 10) D.E. Knuth: Fundamental Algorithms Vol. 1 of the Art of Computer Programming, p. 120, Addison-Wesley, Reading, Mass.(1968).
- 11) 的場, 中北: シンボリック言語によるプログラム教育用システム, 信学誌論 53-C, 1(1970).
- 12) 的場, 家長: 会話形式によるアセンブラ言語教育用システム, 信学誌論 54-C, 6(1971).
- 13) 水島章次: コンピュータ教育用プログラムシステム, 昭 47 情報処理学会大会予稿集 (1972).
- 14) 上野他: 教育用アセンブラ AESOP, 昭 47 情報処理学会大会予稿集 (1972).
- 15) 小島宏行: 教育用マクロアセンブラとプロセッサの一試案, 昭 48 情報処理学会大会予稿集 (1973).

(昭和 50 年 2 月 3 日受付)

付 録

① \$FLN \$FCOM
\$FLOG

\$EALY

② \$HE

YOU ARE WELCOME.
THIS SYSTEM IS THE FCOM VCR#-00 OSAKA UNIV.
THE FCOM IS THE PSEUDO MACHINE FOR EDUCATION.
SO THIS IS VERY SIMPLE AND OLD-FASHIONED COMPUTER SYSTEM.
YOU LOG IN THE FCOM AT THE FOLLOWING TIME.
DATE: 25-SEP-75
TIME: 13:31:47
YOU MAY TYPE A COMMAND AFTER \$.
LET'S GO!

③ \$ED

EDITOR: FCOM VCR#

④ *NEW

ORG 100

LIX 2

⑤ GET #0,X

LEA 10

⑥ EPL **2

MLT M

STA F

FUT F

HLT

A+DEC 10

FINES 1

M+LEC -1

Z+DEC 0

⑦ ENL

TC

⑧ \$LIST

SOURCE PROGRAM LIST FCOM VCR#

ORG 100

LIX 2

GET #0,X

LFA 10

EPL **2

MLT M

STA F

FUT F

HLT

A+DEC 10

FINES 1

M+LEC -1

Z+DEC 0

END

⑨ \$FE

ASSEMBLER: FCOM VCR#

END OF PASS 1

⑩ DO YOU NEED A SYMBOL TABLE LIST BY

LOC MACHINE CODE SYMBOL LIST FCOM VCR#

100

0000111

ORG 100

LIX 2

101

1101010

GET #0,X

LFA 10

102

0140101

EPL **2

MLT M

103

0140101

STA F

FUT F

104

0000100

HLT

105

0000100

A+DEC 10

106

1000001

FINES 1

107

0000000

M+LEC -1

108

0000000

Z+DEC 0

109

0000000

END

END OF PASS 2

BYTES SELECTED: 0000

⑪ DO YOU NEED A SYMBOL TABLE LIST BY

USER SYMBOL TABLE FCOM VCR#

A 100

Z 111

B 109

M 110

END THE ASSEMBLER MODE FCOM VCR#

⑫ \$E 100

⑬ TEST FUN

*1 Ir Acc FCOM VCR# Xr

100 0000111 0000000 0000000

*2 9999 (入カデ-9)

101 1101010 0000000 0000000

⑭ *E 10

*1 0000000

*2 00100010 0000000 0000000

*3 0040101 0000000 0000100

*4 0010100 0000000 0000000

*5 0000000 (出力結果)

106 0000100 0000000 0000000

HALT AT 107

⑮ \$FY

THIS IS THE SYSTEM FCOM VCR#-00 OSAKA UNIV.
YOU LOG ON THE FCOM AT THE FOLLOWING TIME.
DATE: 25-SEP-75
TIME: 13:31:50

ALIAS!

FFATY

⑯ \$FLOG