教育用プログラミング・システム*

原　田　賢　一**

1. はじめに

総合科学として「計算機科学」あるいは「情報科学」といわれるような計算機の専門教育が各所で行なわれるようになってきた一方，従来行なわれていた大学の一般教養課程や専門課程の中でも計算機に関する教育がさかんにとり入れられている．一般に計算機教育といってもその目標には様々あり，大体次のように分けることができる．

（1） 計算機に関する基礎知識の養成　計算機システムの構成と処理能力，アルゴリズムとプログラミング，システムの概念，および利用分野の広さなど計算機についての基礎知識を養い正しい認識を与える．

（2） 計算機を道具として使いこなせるようにするための教育　計算機以外の専門分野の人たちに対して，自分の研究を進めていくうえで計算機を道具として使いこなせるように必要な知識と経験とを与える．

（3） 専門家の教育　計算機自体の開発および計算機の有効な利用法を開発するためにハードウェアおよびソフトウェアの専門家を養成する．

（1）と（2）を目標とした教育の中でも，とくに計算機の働き，アルゴリズム，あるいはプログラミングについて正しい認識と深い理解を与えるには，講義の進行に則した演習と実習を欠かすことはできない．ここでは多人数を対象としたプログラミングの実習に有効なシステムについて考えていくことにする．

2. 教育・実習用のプログラミング・システム

プログラムを処理する方の側からみると，上述の目的のために作られたプログラムは一般に次のような性質をもっている．

（1） プログラムの大部分は数十ステートメントであり，長いものでも300ステートメントぐらいである．実行時間も秒の単位であり，データの量も少ない．（2） 処理件数が多い．（3） 使用される言語は特定のものに限られる．（4） 文法的あるいは論理的なエラーのあるものが多い．（5） ターンアラウンド時間はできるだけ短くしなければならない．

現在広く使用されているソフトウェアは融通性にとんだ処理が可能なこととシステムの資源を最大限有効に利用することを目標として作られ，生産的な処理をするにはすぐれている．このようなシステム（とくに言語プロセッサ）を用いて上述のような性質のジョブを連続処理していくことはできるが，機能が十分すぎてかえって効率の悪い結果を生じることがある．また，使用者の多くが計算機の使い方に不馴れなためで，システムが印刷する各種のメッセージの意味が完全に理解できないこともある．実習に用いるシステムとしては，実習の様子を教員または指導員が適宜知るために，処理記録をファイルしておいて各種の統計資料を出力するような機能を備えている必要がある．これらの理由から，次のような面で教育および実習を主目的としたシステムが開発されている．

2.1 言語プロセッサを含めた実習用のオペレーティング・システム[1)]

学生には指導員が指定した問題についてだけプログラムを作らせるようにする．プロセッサは学生のプログラムを一つのサブルーチンの形にコンパイルする．実行にはいるときは実習用モニタが計算に必要なパラメタをつけてそのサブルーチンを呼ぶ．サブルーチンからコントロールがもどると，実習用モニタはその計算結果を検査する．結果が正しければパラメタを変えてふたたびサブルーチンを呼ぶ．このようなやりとりを何回か繰り返し，パラメタが例外的な値をもっていたときにも正しい処置ができているかどうかなどを調べる．学生のプログラムを処理するたびに実行時間，プログラムの大きさ，結果の正しさ，処理の終了原因などを記録しておき，それらのデータと正しい結果が出るまでの実行回数をもとに実習の成績をつける．これらのデータは報告書作成プログラムによって編集され指導員用のファイルに入れられる．

2.2 既存の言語に対する実習用プロセッサ

多数の小さなプログラムを高速で処理するために作

* Programming Systems for Processing Student Programs, by Ken'ichi Harada (Keio Institute of Information Science)

** 慶応義塾大学情報科学研究所

られたロード実行型のプロセッサで，これについては後に詳しく述べる．

2.3 教育用のプログラミング言語の設計とプロセッサ

なるべく少ない概念を用いて学生が徐々に知識を身につけていけるように言語を設計し[2),3),6)]，TSS の会話モードを利用して自習ができるようにしたシステムである[4)]．このようなシステムが開発される一つの目的には指導員を少なくすることがある。学生自身で疑問な点が解決できるようにていねいなメッセージと簡単な質問応答の機能が用意されている．

高レベルの言語のほかに計算機そのものの働きを理解させるために仮想の計算機を設計して簡単な機械語あるいはアセンブリ言語で実習ができるように開発されたシステムも多数ある[5)]．3. および 4. では慶応義塾大学において実習に使用しているコンパイラとアセンブラについて説明する．

3. 実習用コンパイラ

実習用コンパイラ(WATFOR)[7),8)]は 1965 年夏にカナダの Waterloo 大学で学生実習用に作られた FORTRAN IV コンパイラである．WATFOR コンパイラ†は最初 IBM 7040/44 用に開発されたが，その後価値が認められアメリカでは IBM 360 でも動くものがつくられ，さらにその改訂版である WATFIV[9)] というコンパイラも使用されている．IBM 7040 システムでは WATFOR コンパイラ以外に，ジョブ・モニタのもとに Cobol コンパイラやアセンブラと一緒にぶらさがっているもの (IBFTC) がある．このコンパイラはプログラム単位の処理を行ない，一つのプログラムを処理するまでにはシステム・テープから何回かプログラムを呼ぶ必要がある．作り出されたオブジェクト・プログラムは再配置可能な形になっているので，実行に移るにはローダを通さなければならない．したがって，前述のような多数の小さなプログラムの処理に IBFTC コンパイラを用いると，CPU 時間の大部分は磁気テープの動作待ちに費されてしまう．10 件のジョブを WATFOR コンパイラで処理したときと IBFTC コンパイラで処理したときの比較を**表 1** に示す．IBM 7040 の補助記憶装置は磁気テープしかないために両者の処理時間に大きな相違のあることがわかる．

WATFOR コンパイラについては参考文献 7 で述べられているので，ここではその中からおもな特徴だけを選んで説明をする．

3.1 高速なコンパイル

処理の効率をあげるためにコンパイラはコアに常駐している．処理方式としてはロード実行型を採用し，コア内だけの処理でソース・プログラムから直接機械語に翻訳する．オブジェクト・プログラムを実行している間はコンパイラおよびオペレーティング・システムに記憶保護がかけられるので，システムがこわされることはない．

3.2 診 断

コンパイル時と実行時におけるエラーの診断機能が十分に備わっている．エラーは 3 文字のコードで表示される (そのために使用者はエラー・コードの表を引いてその意味を理解しなければならない)．実行時の

表 1 WATFOR と IBFTC の処理時間の比較

	文の数	副プログラムの個数	コンパイル時間 (秒)		実行時間 (秒)		処理時間 (秒)		印刷された用紙の枚数	
			WATFOR	IBFTC	WATFOR	IBFTC	WATFOR	IBFTC	WATFOR	IBFTC
1	50	0	1.3	34.0	0.95	0.18	2.2	54 0	5	7
2	46	1	1.2	56.0	0 17	0.03	1.4	76.0	3	8
3	50	0	1.3	40.0	0 38	0.20	1.7	59.0	2	5
4	151	1	3.4	93.0	0 67	0.32	4.1	92.0	4	8
5	63	1	1.7	61.0	0 68	0.35	2.4	79.0	3	7
6	72	1	2.0	63 0	0 67	0.37	2.6	81.0	3	7
7	106	4	2.4	118.0	0.38	0.17	2.8	141.0	6	13
8	31	0	0.9	40 0	0 87	0.48	1.7	58.0	2	6
9	35	0	1.0	40.0	0.27	0.20	1.2	58 0	2	6
10	100	2	2.2	115.0	1.20	0.17	3.4	135.0	5	7

(注 1) 処理時間 (IBFTC) の中にはコンパイル時間と実行時間の他にローダによる処理の時間を含む．
(注 2) 印刷用紙の枚数の中には制御カードやコンパイラが出力するメッセージの分も含む．
(注 3) 入出力媒体には磁気テープを使用している．入力テープの作成と出力テープの印刷はオフラインの計算機で行ない，それに費した時間は含まれていない．

† IBM 7040・WATFOR コンパイラは現在早稲田大学，青山学院大学でも使用している．

診断の例としては，値が未定義の変数を参照したときのチェックがある．この場合にはエラー・コード，変数名およびそのような変数を使用した文の位置（文の番号とそれからの相対位置）が表示される．このチェックは配列要素の参照についても行なわれる．このほかに，添字式の値が宣言した大きさをこえてしまったり，副プログラムや関数の引用において引数の型や個数が異なっていたりしたときの診断もなされる．したがって，WATFOR コンパイラは実習用としてだけではなく，大きなプログラムを作るときのデバッグにも有効である．

図1に WATFOR コンパイラにおける記憶装置の使い方を示す．この図では1つのジョブが2つのプログラム単位（AとB）からなっている場合を示している．実行時サブルーチン群には入出力文を処理するためのサブルーチンと基本外部関数のサブルーチンが含まれている．記号表はソース・プログラムで使用した変数名，配列名，副プログラム名，文の番号，定数などを登録しておくためのものである．各要素に対して4語からなる登録項目がつくられ，それらの名前以外に属性を示すビット，コンパイラが文法的なエラーを検出するのに必要なビット，およびデータ領域の割付け番地などを含んでいる．記号表は実行時に論理的なエラーを検出するのにも使用されるので，実行が終了するまで記憶装置に残っている．データ領域はジョブを構成しているすべてのプログラム単位の処理が終わったあとに割り付けられるもので，共通ブロックに属していない変数や配列に対する記憶場所である．その次の領域に共通ブロックに対する記憶場所が割り付けられる．

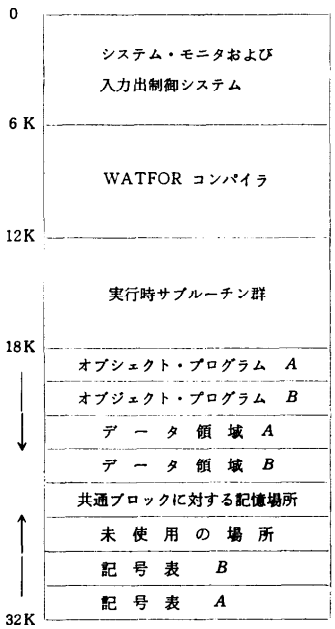

図 1 記憶装置の使い方

慶応大学で使用している WATFOR コンパイラには実習の記録がとれるように多少の修飾がしてある．その中にはプログラム ID 番号，コンパイル時間，実行時間，エラー・コード，使用した記号の個数，実行終了の原因を示すコードなどが含まれる．これらのデータは出力テープに書き出し，オフ・ラインの計算機でカードにせん孔し回収する．このコンパイラは，数値計算，統計計算または一般的な問題をそれぞれ中心とした計算機の入門講座（受講生約400名）と工学部の講座「計算機械および数値計算」（受講生約500名）の実習に使用している．**図2～図5**は昭和46年5～7月における入門講座の実習記録をまとめたものである．図2は1件あたりの平均コンパイル時間，平均実行時間，およびオブジェクト・プログラムの平均語数を示す．図3は処理の内訳，図4はエラーの頻度表，図5はエラーの種類をグループごとにまとめた頻度表である．

実習用コンパイラとしては WATFOR 以外に IBM 7090/94 用に作られた PUFFT (the Purdue University Fast Fortran Translator) が有名である[10,11]．

4. 実習用アセンブラ

実習用アセンブラは工学部管理工学科の「計算機械

COMPILATION TIME	.....	1.478	SEC
EXECUTION TIME	.....	8.333	SEC
TOTAL TIME	.....	9.811	SEC
OBJECT PROGRAM	.....	350	WORDS
DATA STORAGE	.....	363	WORDS
SYMBOL	.....	48	

図 2 ジョブ1件あたりの平均処理時間とオブジェクト・プログラムの平均的な大きさ

CORRECT PROGRAM	.....	4948	
NON CORRECT PROGRAM	.....	6093	
WARNING	.....	12	PERCENTS
FATAL ERROR	.....	60	PERCENTS
RUN TIME ERROR	.....	27	PERCENTS
TOTAL	.....	11041	

図 3 処理したジョブの内訳

```
**********    WATFOR DIAGNOSTICS REPORT   2    **********

FROM 71-05-05 TO 71-07-11

MMO     1392    MIXED MODE // INTEGER WITH REAL
VAO      912    ATTEMPT TO REDEFINE THE TYPE OF A VARIABLE NAME
SXO      868    SYNTAX ERROR -- SEARCHING FOR OPERATOR, NONE FOUND
UVO      767    UNDEFINED VARIABLE
STO      680    UNDEFINED STATEMENT NUMBERS
```

図 4 エラーの頻度表（左からエラー・コード，件数，エラー・コードの意味を示す）

SUBPROGRAMMES	397
SUBSCRIPTS	317
STATEMENTS AND STATEMENT NUMBERS	2391
SUBSCRIPTED VARIABLES	247
SYNTAX ERRORS	994
I/O OPERATIONS	
UNDEFINED VARIABLES	767
VARIABLE NAMES	1326

図 5 エラーの種類をグループ化したときの頻度表

応用」の実習（本特集号“コンピュータに関する専門教育”参照）に用いられているもので，TOSBAC-3400用に作られている[13]．このアセンブラを作ったときの設計目標は次のとおりである．

（1） 処理の高速化

実習ではそれほど高級なことはしないので，アセンブリ言語[12]の仕様は簡単なものにしコアだけでアセンブルを行なうようにする．アセンブラはコア常駐とする．その結果使用者が使える記憶場所は小さくなるが（約3,000語），実習ということを考えればきつい制限にはならない．

（2） 実習記録の集計

各学生のつくったプログラムの性質，エラーの種類，実習回数などを自動的に記録する機能を設ける．これらの情報は実習の指導や方法を考えたり，講義で理解しにくかった点を知ったり，アセンブリ言語やシステムを改良したりするのに役立つ．

（3） 操作の簡略化

計算機の操作は学生自身でやらせることを目標としたために，操作卓のスイッチやボタンはできるだけ触れないでもよいようにし必要な指示や計算機からの表示にはオンラインの入出力タイプライタを用いる．

（4） 共通サブルーチンの作成

入出力や数値変換はほとんどのプログラムで必要となり，機械語でいちいち行なうのは学生にとって相当な負担になる．また，これらが理解できないと実習をはじめられないのでは講義と歩調を合わせて実習を行なうことはできない．そこで，入出力，数値変換，および基本的な関数はサブルーチンとして用意しておき，1つの命令でそれらが使えるようにする．これらについての知識は授業が進んでから種明しをすればよく，はじめのうちは天下り的に使用させる．そうすることによって，固定形の四則演算を教えたときから実習ができるようになる．

（5） アセンブラの保護

学生のつくったプログラムにどのような誤りがあっても次のジョブを処理するのに必要なものだけはこわされないように記憶保護をしておく．ハードウェアの機能として記憶保護とオペレーティング・モードが用意されているので，通訳・実行型のプロセッサにする必要はない．

実習用アセンブラの構成は次のとおりである．

（1） アセンブラ・モニタ

制御カードの解読，実行時間および印刷行数の管理，実習記録の書き出しを行なう．実習中はこのアセンブラしか使わないので，特別な制御カードが出てくるまではこのモニタがジョブを管理する．

（2） アセンブラ

見かけ上1パスでソース・プログラムから機械語への変換を行なう．アセンブル中にその時点で未定義のラベルを参照している命令があると，番地部には未定義の値を入れた機械語をつくりロケーション・カウンタの示す場所に格納する．このとき，その機械語の番地とラベルとを対にして表へ登録しておく．プログラムの読込みが終わると，その表をもとにラベル表を引きながら番地部のうめ込み行なう．

（3） 実行時サブルーチン

カード読込み，印刷，改行制御，各種の数値変換などを行なうサブルーチン群である．学生のプログラムはスレイブ・モードで実行するので，これらのサブルーチンは特権命令の処理プログラムによって間接的につなぎがとられる．

（4） 実習記録の処理プログラム

実習記録は紙テープにせん孔される．これには次の

ような情報が含まれる．

ID 番号，アセンブル時間，実行時間，ソース・プログラムのカード枚数，実行終了原因，使用した印刷用紙の枚数，使用したラベルの個数，文法的なエラーの種類と個数．実習が終わると，この紙テープと処理済みの実習記録のファイルとのつき合わせをしファイルを更新する．このファイルからは個人ごとの実習記録，各回ごとの実習記録について詳細な報告書と要約的な報告書，および実習全体に関する要約的な報告書がとれるようになっている．

図6は今年の5月から7月まで行になわれた実習の処理結果を要約したものである．

<<<<< S U M M A R Y >>>>>

1841 PROGRAMS (*)

	TOTAL	AVERG
ASSEMBLE TIME	666. 51	0. 22
RUN TIME	267. 36	0. 9
TOTAL TIME	934. 26	0. 30
CARDS	175757	95
SYMBOLS	40871	22
USED PAGES	9010	4

END REASON	TOTAL	RATIO TO (*)
0. NORMAL END	992	54 %
1. ASSEMBLE ERROR	450	24 %
2. EXU ERROR OR MEMORY PROTECT	236	13 %
3. TIME LIMIT OVER	101	5 %
4. LINE LIMIT OVER	62	3 %

ERRORS	TOTAL	RATIO TO (**)
CONTROL CARD OR STAT ERROR	36	2 %
NAME ERROR	508	28 %
ILLEGAL OP CODE	177	10 %
ILLEGAL CHARACTER	759	41 %
ADDRESS EXPRESSION ERROR	233	13 %
CONSTANT ERROR	5	0 %
INDEX SPEC ERROR	123	7 %
ORG, EQU ERROR	0	0 %
PROGRAM OVERFLOW	4	0 %
SYMBOL TABLE OVERFLOW	0	0 %
ERROR TOTAL (**)	1845	

図 6 アセンブラによる実習の処理記録

5. おわりに

計算機の教育がさかんになるにつれて，プログラムの実習のために計算機を使用することがますます多くなる．そこで，前にも述べたとおり機能をいくらか削除してもよいから，多数のジョブを効率よく処理し，十分な診断機能を備えたプロセッサの開発が必要になってくる．このようなプロセッサは単に実習用としてだけではなく，一般の使用者にとっても診断機能を利用することによって大いにデバッグの手助けとなる．言語プロセッサを含む実習専用のシステムとしては，使用者のレベルに合わせた適切なメッセージの表示，実習の評価，実習の状態を適宜把握するための資料の作成などの機能を備えていることが望まれる．

最後に，WATFOR コンパイラと実習用アセンブラの整備に協力して下さった慶応義塾大学情報科学研究所の大野義夫氏，小川照易氏，杉田正明氏に深く感謝いたします．

参 考 文 献

1) I. N. Capon: Master-A Subsystem for Processing Student Subprograms, Conference Edition, IFIP World Conf. on Computer Education, p. Ⅲ/167, (1970)
2) L. M. Bohnert: User Oriented Languages for Self Education, Confercence Edition, IFIP World Conf. on Computer Education, p. III/159, (1970)
3) R. R. Fenichel: Design of Languages for Elementary Programming Instruction-Lessons for the TEACH Project, Conference Edition, IFIP World Conf. on Conputer Education, p. Ⅲ/175, (1970)
4) R. R. Fenichel ほか: A Program to Teach Programming, C. ACM, Vol. 13, No. 3, p. 141, (1970, March)
5) 浦昭二ほか: IFIP コンピュータ教育世界会議および欧州におけるコンピュータ教育状況，調査報告書，日本情報処理開発センター，(1970)
6) C. R. Bauer ほか: IITRAN/360 Self-Instructional Manual and Text, Addison Wesley Publishing Company, (1967)
7) P. W. Shantz ほか: WATFOR-The University of Waterloo FORTRAN Ⅳ Compiler, C. ACM, Vol. 10, No. 1, p. 41, (1967, Jan.)
8) P. W. Shantzほか: A Guide to Implementing WATFOR under 7040/44 IBSYS, Univ of Waterloo, (1966, April)
9) Cress ほか: FORTRAN Ⅳ/with WATFOR and WATFIV, Prentice-Hall, (1970)
10) S.Rosen ほか: PUFFT The Purdue University Fast Fortran Translator, C. ACM, Vol. 8, No. 11, p. 661, (1965. Nov.)
11) P. G. Moulton ほか: DITRAN-A Compiler Emphasizing Diagnostics, C. ACM, Vol. 10, No. 1, p. 45, (1967, Jan.)
12) 浦 昭二: アセンブリ言語，培風館，(1970)
13) 原田賢一: 実習用アセンブラ システム，OS シンポジウム報告集，情報処理学会 (1970)

(昭和46年8月16日受付)