

ミニコンのシステム・プログラム

丹羽敏行・本田行雄*・尾形芳邦・甲田有男

電子工学教室
(1973年9月11日受理)

System Programs for Mini-Computer

Toshiyuki NIWA, Yukio HONDA,* Yoshikuni OGATA
and Kunio KODADepartment of Electronics
(Received September 11, 1973)

Software of recent general purpose mini-computer is mainly based upon mass storage devices such as magnetic discs. As we had an opportunity for using a cartridge disc device, we revised and developed our software system.

NIT-FORTRAN-3 compiler has the grammar over the JIS 5000 level and upward compatibility with NIT-FORTRAN-2 in which we can manipulate complex and character data.

NIT-SAP/R assembler generates the relocatable object programs, which are linked and allocated in memory by Linkage-Loader.

NIT-BASIC compiler is a conversational language processor and we can use it as a calculator.

The simulation program for virtual machine is intended to give the students an image of typical computer programming.

1. まえがき

最近の汎用のミニコン・システムでは磁気ディスクなどの大容量外部記憶装置を前提としたものが多くなっている。

筆者らの研究室にも、ミニコンが導入されて3年半ほどになるが、最近磁気ドラムあるいは磁気ディスク記憶装置が、短期間ではあるが使用出来ることになった。

そこで、新しい機器構成に対応したコンパイラなどのシステム・プログラムを、新たに開発したので、その概要を報告する。

2. ハードウェアの構成と仕様

cpuを中心にした筆者らのハードウェア構成を Fig. 1

に示す。ここで、実線で示された部分は既設の機器で、破線の部分は今回システム・プログラムの研究のために一時的に使用した外部記憶装置である。その仕様は次の通り。

<カートリッジ・磁気ディスク装置>

記憶容量	2.5 MW
	200 シリンダ
	800 セクタ
	2 ディスク

平均アクセス時間	35 mS
転送速度	312 KB/S

<磁気ドラム装置>

記憶容量	65 KW
平均アクセス時間	10 mS
転送速度	120 KB/S

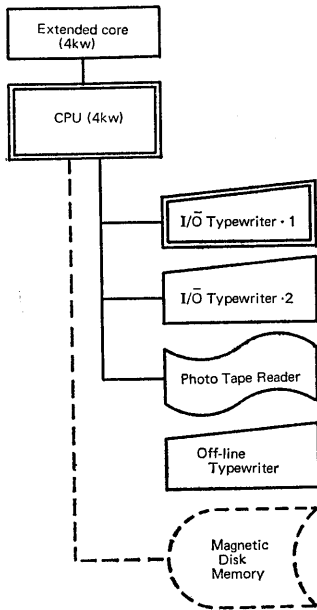


Fig. 1 Block diagram of NIT's Hardware

3. ソフトウェアの構成

Fig. 1 のハードウェア構成に対して準備されたソフトウェアを Fig. 2 に示す。ここで*印を付したものは、今回新たに開発したシステム・プログラムであり、他はすでに報告¹⁾されている。

4. コンパイラ

技術計算用として、バッチ処理用の FROTRAN コンパイラと会話型の BASIC コンパイラを新たに開発した。

4.1 NIT-FORTRAN・3

最近のミニコンのソフトウェア・システムでは磁気ディスクなどの外部記憶装置をベースとしたものが多い。筆者らも、さきに小容量の主記憶のみを用いたミニコン用 FORTRAN コンパイラを実用化し報告^{1) 2) 3)}したが、今回改めて外部記憶装置を使用したよりレベルの高いコンパイラを NIT-FORTRAN・3 と名付け、デバッグもほぼ完了したのでその概要を述べる。

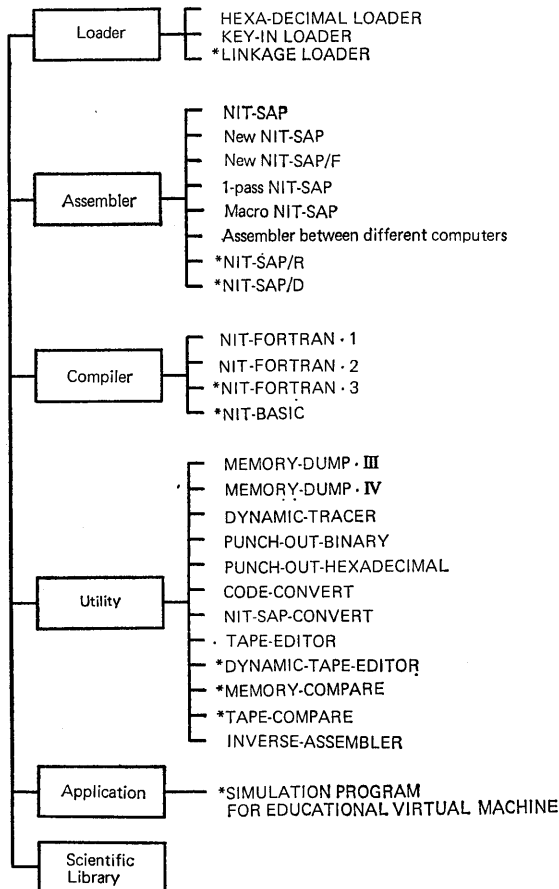


Fig. 2 NIT's software system

NIT-FORTRAN・3 の設計思想は NIT-FORTRAN および NIT-FORTRAN・2 のそれを引継いでおり、上向きの互換性を考慮しながら、教育用という観点から、JIS 5000 を完全に満足するようにした。新 FORTRAN の特徴を Table 1 に示と共に以下に簡単に記す。

(1) 機器構成

最小構成において磁気ドラムあるいは磁気ディスク装置を必要とするが、主記憶の容量および入出力装置の多様性に対して、より柔軟に対処することができる。

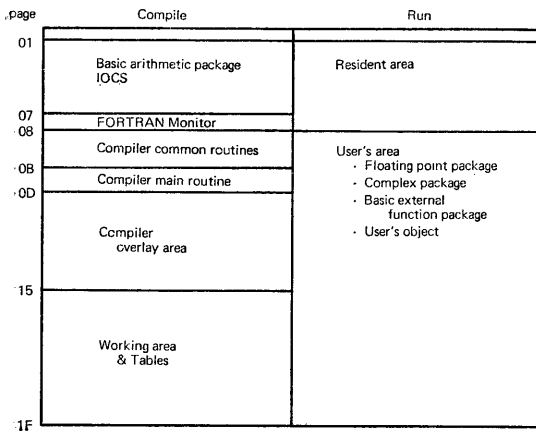


Fig. 3 Memory allocation of NIT-FORTRAN・3

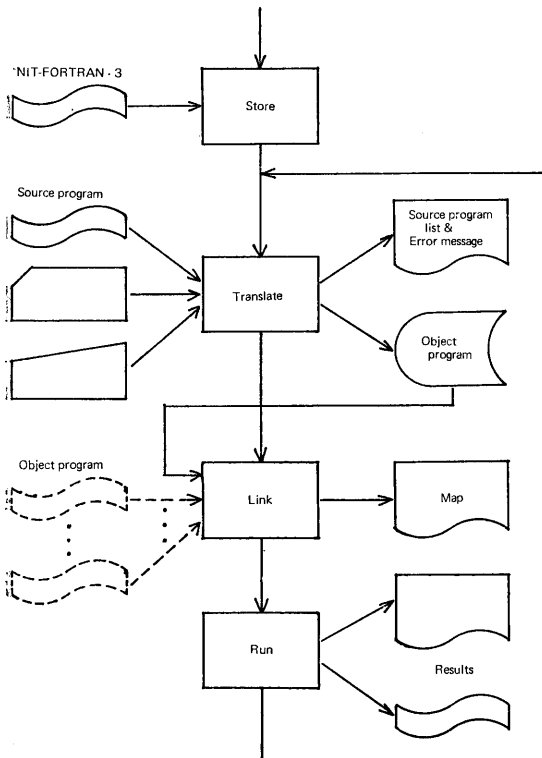


Fig. 4 "Compile and Go" operation of NIT-FORTRAN・3

(2) 処理手順

コントロール・コマンドによる連続バッチ処理が可能で、教育用ジョブが簡単な操作で態率よく処理できる。

(3) コンパイルの方法

旧 FORTRAN ではジョブ単位でしかコンパイルできなかったが、新 FORTRAN ではプログラム単位ごとに独立にコンパイルを行う。

ソース・プログラムは3つのフェイズに分割して処理され、各フェイズの処理ルーチンはオーバーレイの手法により必要に応じて主記憶にロードされる。したがってコンパイル時の常駐領域が少なくすみ、Fig. 3 に示すように各種テーブル類や作業域を大きくとることができる。

また、コンパイル結果の各種の情報の出力がオプションとして用意されており、デバッグおよび他プログラムとのリンクを容易にしている。その操作手順を Fig. 4 に示す。

(4) オブジェクト・プログラムの形

旧 FORTRAN では絶対番地形式のオブジェクト・プログラムが直接生成されるので、コンパイルおよび実行時の能率は非常に良く、また操作性に優れていて小技術計算や教育用に適している。

新 FORTRAN のオブジェクト・プログラムは相対番地形式を採用した。したがって、旧 FORTRAN では不可能であったプログラム単位ごとのコンパイルや、他のプログラムとのリンク、あるいはライブラリの自動コールなどが可能となった。

(5) 文法

補助入出力文と書式なし READ, WRITE 文を除いて完全に JIS 5000 を満足する。また、NIT-FORTRAN・2 の特徴であった複素数演算、文字型データの入出力や演算、およびカナ文字の使用を許し、上向きの互換性を持たせている。

(6) エラーメッセージ

教育用という観点から、エラーメッセージはコンパイル時 116, 実行時24と旧 FORTRAN よりもさらに数を多くし親切なものとした。

(7) 使用結果と問題点

文法のレベル、テーブル類の大きさなど全ての点で旧 FORTRAN を上回っているのは当然としても、アセンブリ・プログラムとのリンクが容易に行えるのは研究用として重宝である。しかし、ミニコン用といえども、JIS 7000 レベルの FORTRAN が常識となりつつあるし、大容量の外部記憶装置をデータ領域として直接使用できないなど、今後の課題は多い。

Table I Comparison between NIT-FORTRAN-2 and NIT-FORTRAN-3

		NIT-FORTRAN-2	NIT-FORTRAN-3
Grammar	Level	JIS 3000	JIS 5000
	Omitted expression	Auxiliary I/O statement Unformatted READ/WRITE statement	Auxiliary I/O statement Unformatted READ/WRITE statement
	Supplemented expression	Type declaration Vertical spacing control Statement function as a real argument READ/WRITE of character data Character-type constant Complex arithmetic expression	READ/WRITE of character data Character-type constant Complex arithmetic expression
Compiler	System storage	Core	Magnetic Drum
	Compiling method	EB object is directly generated	RB object is generated and converted into EB object in link
	Compile unit	1 job	1 program unit
	Object form	Absolute	Relocatable
	Linkage with assembly program	—	available
	Run-time overlay	—	available
	Control command	—	available
	System area	6.4 kw	12 kw
	Resident area in compile	6.4 kw	3 kw
	Resident area in run-time	3.3 kw	1.9 kw
	Object area except tables (with 8kw core) memory	1.6 kw (4.5 kw when EB object is punched out)	4 kw
	Error message	COMPILE 72 RUN 24	COMPILE 116 RUN 24
	Compile & Link Map	—	available
Machine configuration	Minimum	CORE 8 kw, TYP	CORE 8 kw, TYP DRUM 32 kw
	Maximum	CORE 16kw, TYP PTR, CR, LP, HTP	CORE 32 KW, TYP DRUM 131 kw, RTR CR, LP, HTP

4.2 NIT-BASIC

BASIC コンパイラは、タイムシェアリングシステムのために開発された科学技術計算向きの会話型言語で、プログラムはタイプライタのキーボードより直接入力するために使い易い命令形態になっており、また修正も容易である。本コンパイラは、この BASIC コンパイラ

をシングルユーザーを対象として、ミニコン用に修正したもので、次の特徴をもつ。

- a 最少限度の命令数を持つ。
- b 会話型言語の特徴を活かすため、修正機能が豊富である。(文字、ステートメントの修正、ステートメントの挿入、置換えが自由)

- 高度な計算も可能にするため、組込み関数が用意されている。

使用できるステートメントは次の通りである。

```

LET    <変数>=<式>
GOTO  <行番号>
IF    <式1><関係演算子><式2>
THEN  <行番号>
FOR   <単独変数>=<初期値>
TO    <最終値>
NEXT  <単独変数>
READ  <変数>, <変数>, ...
DATA  <定数>, <定数>, ...
PRINT <変数>, <記号>, ...
DIM   <1英文字>, (<整数>), ...
REM   <ストリング>
END
    
```

これらのステートメントによって書かれたプログラムを、キーボードより入力し、計算機と対話しながら修正し、プログラムを完成後すぐ実行できるという手軽さのため、プログラマブルなカリキュレータとして、また初心者のプログラム学習に適している。

5. アセンブラ

従来、筆者らがコンパイラその他のシステム・プロ

ラムを記述するのに用いてきたアセンブラは、IPL から直接入力できる絶対番地形式のオブジェクト・プログラムを生成し、その操作性の良さのため多用してきた。しかし、システム・プログラムが大きくなり、また多くなるにしたがって、記憶装置上でのプログラムの位置が固定されてしまうのは、発展性に乏しくシステム全体の能率を下げることになる。

そこで、新たにリロケートブルなオブジェクト・プログラムを取り扱うアセンブラとローダを開発したので、その操作手順を Fig.5 に示すと共に簡単に述べる。

5.1 NIT-SAP/R

文法その他の仕様は New NIT-SAP とほとんど同じで互換性があるが、次のような特徴を持っている。

- プログラム単位ごとに独立にアセンブルされる。
- したがって、外部変数と共通領域の定義が可能である。
- プログラムの記述は自由欄形式で、
- 6文字までのラベルが許される。

5.2 リンケージ・ローダ

これは、NIT-SAP/R によって生成された相対番地形式の中間言語プログラム(エレメント)を、各エレメント間の対応をつけながら、実行可能な機械語に変換し主記憶上に格納するプログラムである。また、実行時のオーバーレイに対する配慮もなされている。

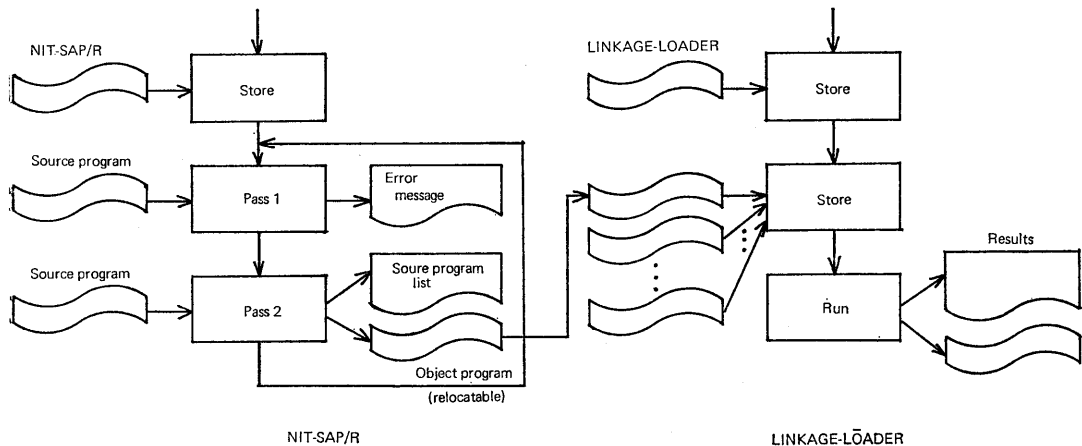


Fig. 5 Assembly operation of NIT-SAP/R and Linkage-Loader

5.3 NIT-SAP/D

これは、主記憶のみを用いた2パス形式のマクロ・アセンブラ New NIT-SAP を磁気ドラムを用いて1パス形式にしたものである。

フェーズ1では、相対番地形式の中間言語を生成し、フェーズ2で機械語に変換する。New NIT-SAP に比較して次の特徴をもっている。

- 1パス形式である。

- アドレス計算式を含んでいる。
- プログラムの記述が自由欄形式である。

6. ユーティリティ・プログラム

ユーティリティ・プログラムはそれ自身で独立したプログラムであり、主にデバッグを行う際に利用される。

6.1 文脈形テープ・エディタ

旧テープ・エディタに比較して、次のような特徴を持

ち、その修正機能の豊富さと使い易さのため、各種ユーティリティの中で最も使用頻度が高い。

- a あらゆる形のソース・プログラムに適用できる。
- b ステートメントの挿入、削除、置換えがポインタの移動によって自由に行える。
- c 修正指示コマンドと挿入されるべき行が明確に区別されていて、修正内容がわかり易い。
- d ポインタはラベルの有無にかかわらず、文字列を指定することによって任意の行に移動できる。また行数だけを指定して相対的なポインタの移動も可能

である。

- e 挿入すべきステートメントの行数を明示する必要がなく、
- f 文字単位の修正が可能で、
- g 何ヶ所もある同一の文字列の置換えが1つの指示で行える。
- h また、修正指示コマンドや文字列の文字数の誤りに対するエラーメッセージがあり、修正誤りを容易にみつかることができる。

修正指示コマンドを Table II に示す。

Table II. Control command of Dynamic TAPE-EDITOR

Command	Meaning
//F, { string $[+n]$ }	Advance the pointer by n lines from { the line pointed by the string } the current line All lines through are punched out.
//S, { string $[+n]$ }	Advance the pointer by n lines from { the line pointed by the string } the current line All lines through are skipped.
//C, m, string-1	Exchange the string by the string-1, which begins at m-th column of the current line.
//X, m, string-1, string-2	Exchange all the string-1 for the string-2 beginning at the m-th column
//N, n	Punch out n feed-codes
///	End of editing

6.2 テープ・チェック・プログラム

これは、磁気ディスクを用いて、紙テープと紙テープの内容を2進のままで比較し、不一致な点でその内容を印字して一時停止するプログラムである。システム・テープをコピーした場合などに必要となる。

6.3 メモリ・チェック・プログラム

旧メモリ・チェック・プログラムの印字形式を改良し、別のプログラム単位を連続してチェックすることができるようにしたプログラムで、用紙1枚に250個所のテープとメモリの不一致個所を印字することができる。システム・プログラムをパンチ出力した場合や、プログラムの実行の前後における主記憶の内容の違いについて知りたいときなどに使用する。

6.4 メモリ・ダンプIV

これは、主記憶の内容を16進で印字すると共に、その機械語に対応するニーモニック表示をも並べて印字するプログラムである。用紙1枚に128語分を印字する。

7. アプリケーション・プログラム

筆者らの使用しているミニコンは、命令の構成や番地参照の方法に欠点や複雑すぎる点があり、機械語レベルの教材用として不適當な面がある。

そこで、教材用として仮想のミニコンの仕様を定め、そのシミュレーションを行うプログラムを作成した。仮想ミニコンのアドレス参照命令のビット構成と修飾方法を Fig. 6 に示すと共にその特徴を次に述べる。

- a 命令体系が整理され命令数も ĆKITAC-4300 よりも多い。
- b マイクロ命令がある。

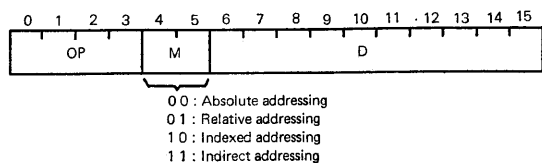


Fig. 6 Memory reference instruction format of virtual machine

- c ダイナミック・トレーサの機能を持っていて、デバッグやプログラムの流れを理解するのに便利である。
- d 番地参照の方法が簡単でわかりやすい。
- e 入出力や浮動小数点演算など豊富なサブルーチンが用意されている。

終りに、ハード、ソフト共に種々便宜をはかっていたいた沖電気工業株式会社関係各位、ならびにプログラミングを手伝っていただいた卒研究生諸君、また日頃適切なご助言をいただく名古屋大学理学部近藤一郎助教授の方々に深く感謝いたします。

8. むすび

以上、この1年間に新たに開発したソフトウェアの主なものを以前のものと比較してみたが、主記憶のみを対象としたソフトウェア・システムの限界を強く感じる。今後ミニコンにも磁気ディスク、磁気ドラム、カセット磁気テープなどの外部記憶装置が一般に用いられるようになることを考えれば、ミニコンのソフトウェア・システムも根本から考えなおすべきではないかと思われる。

文 献

- 1) 丹羽・本田：名工大学報 24 (1972) p. 285
- 2) 丹羽・本田：名工大学報 23 (1971) p. 349
- 3) 丹羽・本田：昭47電学会東海支連大論文集
- 4) 丹羽・本田：昭48電学会東海支連大論文集
- 5) B. Soucek: "Minicomputers in Data Processing and Simulation", John Wiley and Sons, Inc. (1972)