

## 7 エディタと言語プロセッサの結合

東京大学工学部 武市正人

1. ミニコンピュータは、(単独に)パーソナルコンピュータとして、簡単な計算を行うにも便利なものである。あまり計算量、データの量が大きくないプログラムは、計算センターに依頼する手間と、ターンアラウンドタイムを考慮すると、ミニコンピュータで計算する方が能率が良い。

ミニコンピュータに高速の入出力装置あるいは大容量の補助記憶装置が付属している場合はともかく、光電式紙テープ読取装置(PTR)、タイプライタとそれに付いている低速紙テープ穿孔機程度しか付けていない(最小)構成においては、ソースプログラムの作成、修正は容易でなく、また、紙テープの出力もできるだけ避けたいものである。

エディタは、ソースプログラム作成の労力を軽減するために、ミニコンピュータには不可欠なソフトウェアの一つである。そのプログラムを処理する言語プロセッサ(アセンブラ、コンパイラ等)も、途中の結果を紙テープに出力することのない、いわゆる1パス方式が望ましい。言語プロセッサプログラムの格納も一度だけにしたい。つまり、それらは常駐型であるのが望ましい。

エディタを使ってソースプログラムを作成することは比較的容易になった。1パス常駐型の言語プロセッサにより、アセンブルコンパイルも短時間でできる。しかし、ソースプログラムにあるかも知れない(文法上の)誤りは、ソースプログラムで修正するのが普通である。コアメモリにエディタと言語プロセッサが同居していて、コアメモリ内につくられるエディタのファイルを直接言語プロセッサに引渡し、文法上の誤りを無くしてから(必要ならば)紙テープに出力することにより、この手間を省くことができるであろう。

この目的のためには、エディタ、言語プロセッサともに、適切な機能を備え、しかもコンパクトでなければならぬ。ここでは、東京大学森口研究室で開発されたEDITOR-ASSEMBLER、EDITOR-FORTRANについて紹介する。

計算機は、松下通信工業のMACC 7/S 4K語および12K語(1語は16bits)で、入出力機器は上記の(最小)構成である。

### 2. EDITOR, ASSEMBLER, FORTRAN

EDITORはbit\*に紹介されている文脈エディタである。EDITORの占有語数は約500語、コマンドで行内文字列の置きかえ機能を追加して約600語。

ASSEMBLERは、メーカーから提供されたアセンブラと同様の機能を有するコンパクトなもので、占有語数は約750語。

FORTRANは、4K語用常駐1パスのコンパイラ(インタプリタ)を8K語用に拡張したものをを用いる。占有語数は約3100語。

### 3. EDITOR-ASSEMBLER

EDITOR, ASSEMBLERは図1のように配置される。作業用領域にエディタのファイル、アセンブルの際の

\* 和田英一、「ミニエディタ」、臨時増刊bitミニコン、1971年12月(共立出版)pp. 40-47.

ネームテーブル、アセンブルされたプログラムがつけられる。

EDITORからASSEMBLERに制御を移し、エディタのファイルからソースプログラムのテキストを取り出すコマンドとして、>A)を用いる。

#### 4. EDITOR - FORTRAN

4K FORTRANコンパイラの作業領域を8K語増やして12K語用とすると、プログラム領域4K語のプログラムが実行可能である。FORTRANの文で約800~1000ステップのプログラムが実行できるが、このように大きいプログラムは、基礎となる4K FORTRANのコンパイラ(実行はインタプリティブに行われる)の構成からみても、このコンパイラで処理をするのは得策ではないであろう。4K語用FORTRANの作業用領域ではもの足りないので、作業用領域を4K語増やすことにし、あとの4K語をEDITORおよびそのファイルの領域とする。(図9)

EDITORからFORTRANへ制御を移すのは、EDITOR-ASSEMBLERの場合と同様に、コマンド >A)で行う。翻訳終了時、あるいはファイルのテキストが不足しているような場合にはEDITORに制御が戻される。プログラムの実行は、>J)によって再びFORTRANに制御が移され、終了時にはEDITORに戻る。制御の遷移は図8に示す。

#### 5. エディタと言語プロセッサの結合

エディタはその目的として、紙テープにファイルを出力する機能をもつ。EDITORではコマンド >E)によって指示される。

言語プロセッサは、紙テープから文字を読む。

中間媒体の紙テープを用いず、言語プロセッサの文字入力ルーチンは、エディタの文字出力ルーチンから文字を受けとればよい。

この方法で結合することは、エディタの文字出力ルーチン、言語プロセッサの文字入力ルーチンを変更することにより、簡単に行うことができる。

#### 6. エラーメッセージの照会 (inquiry)

FORTRANでは、エラーメッセージは、番号で示され、表を参照することによりその内容を知ることができる。一方、文脈エディタであるEDITORには、文字列一致の条件で、行を捜索する機能(コマンド >F string)が備っている。この機能を有効に利用して、エラーの内容を必要に応じて、照会できるようにしたい。

EDITORは、予め指定されたエラーメッセージのファイルと、ソースプログラムテキストのファイルの2つをもつ。EDITORのF機能をエラーメッセージのファイルに利用する。これは、ファイルのポインタの組を個々に用意することにより可能である。

エラーメッセージの照会にはコマンド >? error no または >! error no.を用いる。(又4)

通常は、ソースプログラムテキストのファイルを扱い、コマンド >?, >!に対しては、エラーメッセージのファイルを扱う。エラーメッセージの印刷が終わった際には、ソースプログラムテキストのファイルを扱う状態に復帰する。

7. ミニコンピュータで、インタラクティブにプログラムの作製計算を行うには、他のプログラム言語（例えば BASIC）が一般的であろう。とくに、デバッグが容易に、短時間で出来ることが要求される。ここでは、既存の言語プロセッサを用いて、その一つの方法を示した。計算実行時のデバッグ機能も必要であるが、最小構成のミニコンピュータにどの程度の機能を備えるかは検討の余地がある。

なお、ここに登場した個々のソフトウェアは、和田英一、算捷彦、辻尚史各氏および武市によって作成されたものである。

図 1

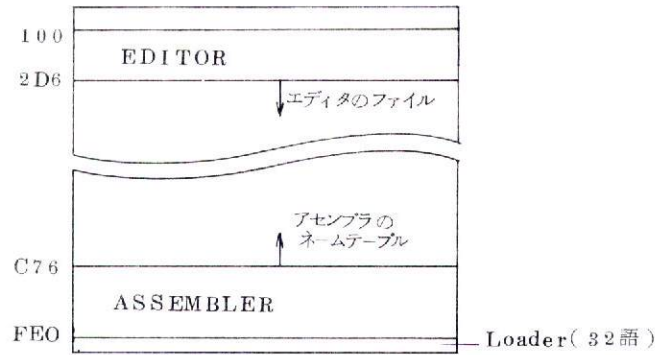


図 2

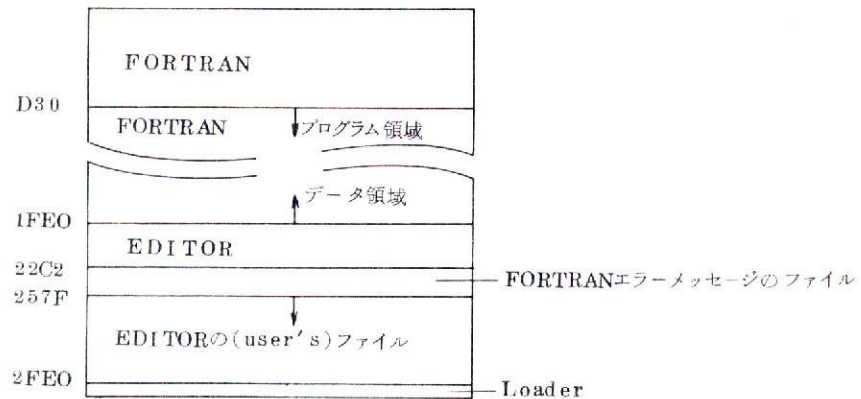
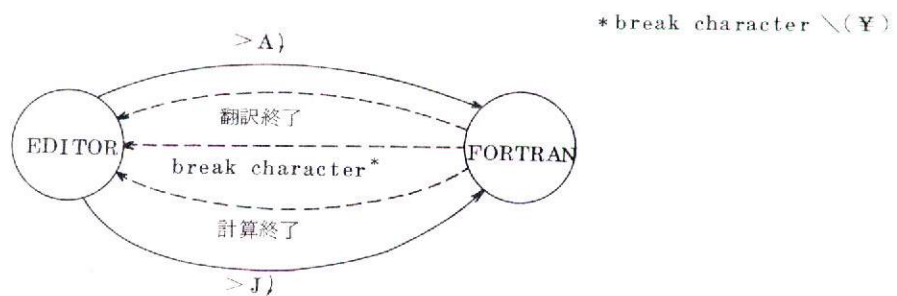


図 3



>Q

>I

```
READ(1,100) A,B;
100 FORMAT(2F12.5);
S=A+B;
D=A-B;
I=A**20;
WRITE(2,200) A,B S,D,I;
200 FORMAT(//2E15.7//2E15.7,17);
STOP;
END;
@
  3.14159    2.71828
```

>A

? 008 006

@ 01E 00A 001

>?008

008 ERROR IN WRITE STATEMENT

>T

```
READ(1,100) A,B;
>FWRITE
WRITE(2,200) A,B S,D,I;
>G:B S:B,S;
WRITE(2,200) A,B,S,D,I;
>A
```

@ 023 00A 000

>J

! 103 016

>!103

103 INTEGER VALUE > 32767

>T

```
READ(1,100) A,B;
>FI=
I=A**20;
>RI=A**2;
>A
```

@ 023 00A 000

>J

```
0.3141589E+01  0.2718279E+01
0.5859869E+01  0.4233098E+00    9
```

/

>E

图 4