

U 1 1 0 0 用エディターの開発

工学部 宇都 幸一・栗須 正登

1. まえがき

近年大型からマイコンに至るまで、エディターは一つのシステムプログラムとして計算機に組み込まれつつある。エディターとファイルの利用によってユーザプログラムの開発は容易となり、ユーザは計算機と会話を行ないながらプログラムの開発が可能となった。本学においてもM180II-AD TSSによるエディター利用は各種言語の教育に大いに役立っており、教育やプログラム開発に欠かせないものとなっている。しかしながら全ての計算機に優れたエディターが提供されているわけではなく、センターに導入されたU1100用エディターはユーザーの使用に対して不十分なものであった。U1100は本来実験などから得られるアナログデータをA-D変換し、以後ユーザプログラムにより処理を行うものであり、その使用形態は会話形式が望ましい。よって他の計算機にも移植可能であり、またよりすぐれたエディターをユーザがFORTRAN言語による開発可能なU1100用エディターの開発を行い、結果を得たのでここに報告する。

2. エディターとFORTRAN言語

エディターはファイルにおさめられているプログラムテキストの行や文字の検索、修正、挿入、削除を行い、その結果を同じファイルや新しいファイルに格納する機能をもっており、データ検索の一種と考えられる。よってエディターのアルゴリズムはデータ処理を研究として行う場合大きな手段となりうる。データ検索を行うプログラムを開発する言語としてはPLI, PASCAL等が、その言語の性質上すぐれているが、いまだに一般ユーザにはあまり使用されていないのが現状である。よってデータ検索においては多くの欠点をもつけれども、その普及性などを考えればFORTRANによる開発は一般ユーザにとって適していると考えられる。よってまずFORTRANによるエディターに使用した文字処理についてのべ、次にエディターの速応性に関して重要なランダムファイルの使用およびその動きについてのべ最後にエディターの使用例についてのべる。

2.2 FORTRANによる文字処理

今回のエディターに使用されたコマンド形式は次の二種がある。

| (a) 形式 | 例 |
|-------------------------|-------------------|
| C1 n1 n2 / P1 /A | F 10 100 / END /A |
| ここで | C 10 / A/AB/ |
| C1 コマンド (F, C, D, L, S) | D 10 100 |
| n 1, n 2 行番号 (10, 100) | |
| p 1 文字列 (END, A, AB) | |
| A 付加機能 (全て) | |

研究報告

(b) 形式

例

n1 STATEMENT

31

GO TO 300

ここで

n 1 行番号 (31)

各命令のコマンド、行番号の区切りはブランク () を使用しており、文字例および付加機能の区切りにはスラッシュ (/) を使用している。

(a)形式のコマンド処理を示すと次の様になる。

2.2.1 コマンドの処理

```
1      DIMENSION I(80)
2      DATA LF,LC,LD,LL,LS/1HF,1HC,1HD,1HL,1HS/
3      DATA LB/1H /
      .
      .
4      READ(5,100)I
      100 FORMAT(80A1)
      .
      .
5      IF(I(1).EQ.LF)GO TO ---
6      IF(I(1).EQ.LC)GO TO ---
      .
      .
```

ステートメントの意味

例

- 1 I (8 0) コマンド文字列 (F 10 100 /END/A)
- 2 LF, ~LSにコマンド文字F, ~Sを定義
- 3 LBに) を定義 ...) (ブランク)
- 4 コマンド文字列の入力
- 5, 6 入力されたコマンド I (1)がどのコマンドかを判定する

2.2.2 行番号の処理

```
1      DIMENSION I(80),NO(10),NS(2),NE(2),NT(2)
2      DATA NO/1H1,LH2, . . . . . 1H9,1H0/
3      DATA LB/1H /
4      DATA LSRA/1H//
      .
      .
      NT(1)=0
      NT(2)=0
      L=0
      DO 1 K=2,72
      KP=K+1
5      IF(I(KP).EQ.LSRA)GO TO 100
6      IF(I(K).EQ.LB.AND.I(KP).NE.LB)L=L+1
7      IF(I(K).EQ.LB.AND.I(KP).NE.LB)NS(L)=KP
8      IF(I(K).NE.LB.AND.I(KP).EQ.LB)NE(L)=K
1 CONTINUE
```

```

100 CONTINUE
    DO 2 M=1,L
      NSS=NS(M)
      NEE=NE(M)
      DO 3 NN=NSS,NEE
        DO 4 K=1,10
9          IF(I(NN).EQ.NO(K))GO TO 5
        4 CONTINUE
          :
10         5 IF(K.EQ.10)K=0
12         NT(L)=NT(L)+K*10**(NEE-NN)
          3 CONTINUE
          2 CONTINUE

```

ステートメントの意味

- 1 NO (10) 整数文字列
- NS (2) 行番号初めのけた
- NE (2) " 最後のけた
- NT (2) 行番号の数字
- 2 NO (1) ~ NO (10) に整数文字を定義
- 3, 4 LB, LSRAにプラック, /文字を定義
- 5 次の文字が / 文字なら次には行番なし
- 6 行番号の個数カウント (L)
- 7 行番号の最初の桁位置 (NS)
- 8 行番号の最後の桁位置 (NE)
- 9 I (NN) がどの整数文字か判定
- 10 K=10 (I (NN) = NO (10)) ときの判定
- 12 行番号の整数への変換

注 コマンド文字列の行番号は文字として入力されており、この文字を数字として変換しなければならない

2.2.3 文字列の格納 (メモリー上へ)

```

1     DIMENSION I(80),I1(40),I2(40),NM(4),NL(2)
2     DATA ISRA/1H//
      NN=0
      DO 1 K=1,72
3     IF(II(K).EQ.ISRA)NN=NN+1
4     IF(II(K).EQ.ISRA)NM(NN)=K
      1 CONTINUE
5     NN=NN-1
      DO 2 M=1,NN
6     LL=NM(M+1)-NM(M)-1
      NL(M)=LL

```

研究報告

```
      DO 3 L=1,LL
      LLL=NM(M)
7      LLLM=LLL+L
      IF(M.EQ.1) I1(L)=II(LLLM)
8      IF(M.EQ.2) I2(L)=II(LLLM)
9      3 CONTINUE
      2 CONTINUE
```

ステートメントの意味

- | | | |
|---|---------------------------|------------------|
| 1 | I 1 (4 0) , I 2 (4 0) | 文字列 1 , および文字列 2 |
| | NL (2) | 文字列の長さ |
| | NM (4) | スラッシュ位置 |
| 2 | I S R A に / 文字定義 | |
| 3 | / 文字のカウンタ | |
| 4 | / 文字の位置 | |
| 5 | 文字列の個数 | |
| 6 | 文字列の長さ | |
| 7 | 文字の位置 | |
| 8 | I 1 に 1 番目の文字列代入 | |
| 9 | I 2 に 2 番目の文字列代入 | |

2.3 ランダムファイルと検索

エディターは一般に会話形式で使用され、使用者の入力に対して、即時に出力が出来なければならない。U1100ではプログラムテキストはシーケンシャルファイルに基づいたディスクファイルに格納されているので、このテキストを全てメモリー上に記憶することが可能であれば、処理速度は早いですが、U1100のようなミニコンでは不可能であり、かつ大型計算機でも実用的ではない。エディターはテキストの行や行の内容を検索し処理するものであり、また検索される行はランダムであるためそのままシーケンシャルファイルを使用することは、ファイルの入出力時間が長くなり実用的ではない。よってこのテキストを一時ランダムファイルに書きこんで処理を行った(U1100では1 byte 単位のランダムアクセスはFORTRAN言語上不可能なのでランダムファイルはシーケンシャルファイルの2倍の容量をとり2 byte 1文字の情報を書き込んだ)。またランダムファイルの番地とテキストの行番号はメモリー上にもうけた。以下にファイルおよび番地の動きを示す。

2.3.1 シーケンシャルファイルよりランダムファイル

2.3.1 シーケンシャルファイルよりランダムファイル

```

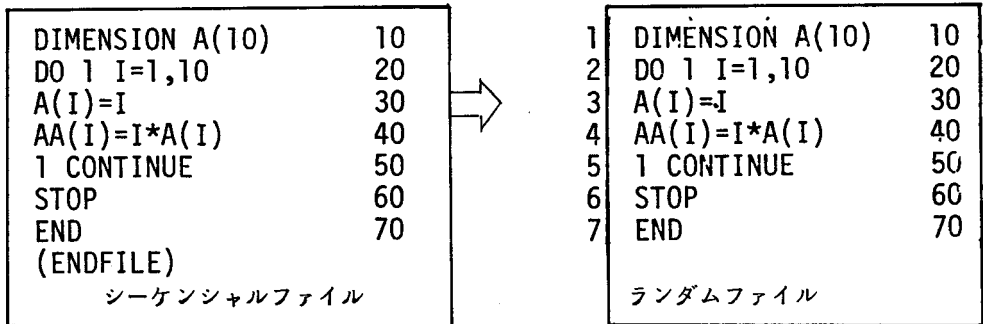
1      DIMENSION II(80)
      :
      DO 1 K=1,1000
2      READ(88,100,END=2)II
100  FORMAT(80A1)
3      CALL PUT(89,II,K):: WRITE(89'K)II
      1 CONTINUE
    
```

ステートメントの意味

- 1 II (80) テキストの文字列
- 2 シーケンシャルファイルから文字列の入力
- 3 ランダムファイルへの出力

注 88…シーケンシャルファイルの機番, 89…ランダムファイルの機番
CALL PUTがない場合にはWRITE(89'K) IIを使用すればよい

ファイルの動き



注 シーケンシャルファイルにはENDFILEが記憶されており、テキストの最後を知ることが出来る。ランダムファイルはない。また今回の場合シーケンシャルファイルのレコードは80, ランダムファイルは160 byteである。

2.3.2 行番号の挿入

51 WRITE(6,*)I,A(I) によるファイルの動き

意味 : 行番号51に以下のステートメントを書け

| | K | N(K,1) | N(K,2) | TEXT(FILE) | |
|-------------|---|--------|--------|-------------------|----|
| 実 行 前 | 1 | 10 | 1 | 1 DIMENSION A(10) | 10 |
| | 2 | 20 | 2 | 2 DO 1 I=1,10 | 20 |
| | 3 | 30 | 3 | 3 A(I)=I | 30 |
| | 4 | 40 | 4 | 4 AA(I)=I*A(I) | 40 |
| | 5 | 50 | 5 | 5 1 CONTINUE | 50 |
| | 6 | 60 | 6 | 6 STOP | 60 |
| | 7 | 70 | 7 | 7 END | 70 |



研究報告

| | K | N(K,1) | N(K,2) | TEXT(FILE) | |
|---|---|--------|--------|--------------------|----|
| 実 | 1 | 10 | 1 | 1 DIMENSION A(10) | 10 |
| | 2 | 20 | 2 | 2 DO 1 I=1,10 | 20 |
| | 3 | 30 | 3 | 3 A(I)=I | 30 |
| 行 | 4 | 40 | 4 | 4 AA(I)=I*A(I) | 40 |
| | 5 | 50 | 5 | 5 1 CONTINUE | 50 |
| | 6 | 51 | 8 | 6 STOP | 60 |
| 後 | 7 | 60 | 6 | 7 END | 70 |
| | 8 | 70 | 7 | 8 WRITE(6,*)I,A(I) | 51 |

ここで

K テキスト順番
N (K, 1) 行番号
N (K, 2) ファイルの番地

である。

処理の手順

- 1) 行番号 5 1 を数字化
- 2) N (K, 1) を調べ K を求める
- 3) K が求まらない場合には N (K, 1) と 5 1 の大小関係よりこのステートメントの順番を定め、以降の順番を決めファイルの最後にテキストを加える
- 3') K が求まるならばその番地 N (K, 2) にステートメントを書く (以降のテキスト内容は消される)

2.3.3 文字列の変換

C 40 /AA/A/ によるファイルの動き

意味 : 行番号 4 0 のテキスト内の A 文字を AA に変換する

4 AA(I)=I*A(I) 40 ⇒ 4 A(I)=I*A(I)

処理の手順

- 1) 行番号 4 0 を数字化
- 2) N (K, 1) を調べ K を求める
- 3) K から N (K, 2) を求めそのテキストから文字 AA を A に書きかえ、以後の文字列を左に 1 桁ずつずらす (ただし行番号はずらさない)

2.3.4 行の削除

D 40 によるファイルの動き

意味 : 行番号を削除する

| K | N(K,1) | N(K,2) |
|---|--------|--------|
| 1 | 10 | 1 |
| 2 | 20 | 2 |
| 3 | 30 | 3 |
| 4 | 40 | 4 |
| 5 | 50 | 5 |
| 6 | 60 | 6 |
| 7 | 70 | 7 |

⇒

| K | N(K,1) | N(K,2) |
|---|--------|--------|
| 1 | 10 | 1 |
| 2 | 20 | 2 |
| 3 | 30 | 3 |
| 4 | 50 | 5 |
| 5 | 51 | 8 |
| 6 | 60 | 6 |

処理の手順

- 1) 行番号40の數字化
 - 2) N(K, 1)を調べ、Kを求める
 - 3) 削除すべきKが求まるとそれ以後の順番を1つつ減少させる
- よってこの場合にはファイルの動きはなく、ただ行番号、番地を処理するだけである。

2.3.5 格納

S によるファイルの動き

意味 : ランダムファイルよりシーケンシャルファイルへの格納

| K | N(K,1) | N(K,2) | | |
|---|--------|--------|--------------------|----|
| 1 | 10 | 1 | 1 DIMENSION A(10) | 10 |
| 2 | 20 | 2 | 2 DO 1 I=1,10 | 20 |
| 3 | 30 | 3 | 3 A(I)=I | 30 |
| 4 | 50 | 5 | 4 A(I)=I*A(I) | 40 |
| 5 | 51 | 8 | 5 1 CONTINUE | 50 |
| 6 | 60 | 6 | 6 STOP | 60 |
| 7 | 70 | 7 | 7 END | 70 |
| | | | 8 WRITE(6,*)I,A(I) | 51 |



```

DIMENSION A(10)           10
DO 1 I=1,10               20
A(I)=I                    30
1 CONTINUE                50
WRITE(6,*)I,A(I)         51
STOP                      60
END                        70
(ENDFILE)
    
```

処理の手順

- 1) N(K, 2)のKを1からテキストの最大値まで変化させて、ランダムファイルのN(K, 2)番地のテキストを読みこみ、シーケンシャルファイルに書き込んでいく。
- 2) 1)の処理が終わったらENDFILEを書きこむ。

研究報告

2.4 機能

表1にコマンドの機能を示す。

表1

| コマンド | 機能 |
|----------------------|--|
| L n1 (n2) | 行番号 n1 [~n2までの] リストアップ |
| C n1 / P1 / P2 / | " の文字列 P1 を P2 に変換する |
| F n1 (n2) / P1 / [A] | " [~n2までの] テキスト中に文字 P1 を含む [全ての] テキストをリストアップ |
| D n1 (n2) | 行番号 n1 [~n2までの] のテキスト削除 |
| n1 STATEMENT | に以下の STATEMENT を書く, n1 がなければ新しく加える |
| S | 編集したテキストを新しいファイルに格納する |

注 表中の [] のコマンド文字がない場合には機能での [] の文字をのぞいた機能となる

注 F n1 / P1 / A は行番号 n1 に文字列 P1 が2個以上ある場合意味がある。

2.5 使用例

U1100でこのエディタを機動させるためには // EDIT と入力する

// EDIT ファイル名 (COMPEL) のテキスト編集 (リストアップ, 修正, 挿入, 削除, 検索) を行ない結果を別のファイル名 (FILEI) に格納する。

L 10 120 (リストアップ : 行番号10から120まで)

```

00010      DIMENSION IA(128)
00020      REWIND 11
00030      READ(11) IA
00040      WRITE(6,60) (IA(I),I=1,16)
00050      60  FORMAT(1H ,8[6,4X,8A2)
00060      DO 100 J=1,4
00070      READ(11) IA
00080      WRITE(6,61) IA
00090      61  FORMAT(1X,16I6)
00100     100  CONTINUE
00110      STOP
00120      END
  
```

L 50 (リストアップ : 行番号50のみ)

```
00050      60  FORMAT(1H ,8[6,4X,8A2)
```

C 40 /16/128/ (修正 : 行番号40のステートメントの文字列16を128に)

```
00040      WRITE(6,60) (IA(I),I=1,128) ...修正結果
```


55 A(17)=12345 (挿入 : 行番号55の相当する位置に)

L 50 60 (リストアップ)

00050 60 FORMAT(1H ,8I6,4X,8A2)
00055 A(17)=12345
00060 D0 100 J=1,4

挿入結果確認

C 55 /A/IA/ (修正)

00055 IA(17)=12345 ... 修正結果

56 IA(18)=!!!0 --CAN (CAN) キー : キャンセルキー ...キーイン・ミス時

56 IA(18)=11111

57 IA(199-2222) (挿入)

58 IA(20)=3333

L 50 60 (リストアップ)

00050 60 FORMAT(1H ,8I6,4X,8A2)
00055 IA(17)=12345
00056 IA(18)=11111
00057 IA(199-2222
00058 IA(20)=3333
00060 D0 100 J=1,4

挿入・結果・確認

D 55 58 (削除 : 行番号55から58までのステートメント)

L 50 60 (リストアップ)

00050 60 FORMAT(1H ,8I6,4X,8A2)
00060 D0 100 J=1,4 } 削除結果確認

F 10 120 /IA/ (検索 : 行番号10から120までのステートメントについて
文字列IAが存在する最初のステートメントの発見)

00010 DIMENSION IA(128)

F 10 120 /IA/A (検索 : 行番号10から120までのステートメントについて
文字列IAが存在するステートメントをすべてリストアップする)

00010 DIMENSION IA(128)
00030 READ(11) IA
00040 WRITE(6,60) (IA(I), I=1,128)
00070 READ(11) IA
00080 WRITE(6,61) IA

S (テキスト編集結果をファイル名FILEIに格納する)
J STOP 0000

研究報告

3. 考察

今回開発したエディターはRENUMやMOVE, COPYなどの機能がないためまだ完全とはいえないが、入力および出力がタイプライタであるため、一応これだけの機能で十分なテキスト編集が可能であり、また命令に対する速応性も、ほぼ満足できるものである。またこのエディターはFORTRAN言語で作成されており、他の機能を加えることも容易にでき、かつ他のシステムに容易にエディターの移植が可能である。

4. あとがき

このエディターの開発に伴ない、種々の処理を理解することができた。特に開発当初はシンケンシャルファイルでテキストの検索を行っていたため、1つのコマンドに対する応答が非常に遅く、実用性を欠いていたが、ランダムファイルを用いることにより解決することができた。またテキストの削除ではテキストの番地をメモリーに記憶することでファイルのI/Oを無くすことができた。

最後にこの研究に対して多くの助言をいただいた野崎, 阪上氏をはじめとし、センター各位に謝意を表します。この研究について計算機はU1100, FACOM270-20を使用した。

参考文献

- 1 PANAFACOM FORTRAN-D 説明書
- 2 PANAFACOM コマンド使用手引書