

シヨートノート

マクロコマンドによるソースプログラム生成方式
の簡易形情報検索システム†

清野 武竹 池田 克夫†† 島崎 眞昭†††

マクロコマンドを利用して、情報検索のためのプログラムを簡易に生成する方式について述べている。その特徴はつぎのとおりである。

- i) 単一コマンドを与えるのみで情報検索できる。
 - ii) 検索条件の指定はコマンドのパラメタとして、FORTRAN の任意の論理式を用いて行うことができる。
 - iii) システムはわずかの労力で簡単に開発できる。
- システムは検索条件を指定する論理関数サブプログラムを用いるが、これは検索の際、コマンドパラメタより、テキストエディタを用い、ソースプログラムが生成される。システムはオペレータコマンドプロセッサ、テキストエディタ、コンパイラ等を有機的に結合、活用している。

1. はじめに

大規模情報検索システムはその機能、汎用性の点で有用であるが、一般にシステムの設計、開発、維持管理に多大の時間と労力を必要とする。一方簡易形の情報検索システムであっても、1) 操作性に優れ、2) 開発が容易である、ならば、その価値は十分高いと考えられる。我々はマクロコマンドによるソースプログラム生成方式の簡易形情報検索システムを開発し、使用しているので、その構成法について報告する。特徴的なこととして次のことが挙げられる。

(1) 検索操作としては端末よりコマンドを一行投入するのみでよい。

(2) 検索条件は FORTRAN 文法に従う任意の論理式の形で指定できる。

(3) 検索システムの開発に要した労力は 0.5 人日程度である。

検索システムの方式はマクロコマンドによるソース

プログラム生成方式であり、システム作成の手法の面からみると、1) オペレータコマンド言語とくにそのマクロ機能、2) テキストエディタ、3) FORTRAN コンパイラ、4) 結合編集機能、5) タスク実行機能を有機的に結合している。とくに、コマンド言語のマクロ機能が巧妙に利用されている。

2. データベースの形式と検索について

我々が対象としたデータベースは比較的単純であるが一般なもので、例としてよく用いられる人事管理のデータベースに似ている。ファイルはすべて同一の形式をもつレコードからなり、各レコードは幾つかの属性をもっている。

FIELD 1	FIELD 2	FIELD 3	FIELD #
---------	---------	---------	---------

図 1 レコードの形式

Fig. 1 Record format.

属性が検索条件を満たすレコードのみ、レコード番号ないしその属性を指定するファイルに出力するとする。検索条件はたとえば

- (FIELD (1). GT. 20). AND.
- (FIELD (2). EQ. 'A '). OR.
- (FIELD (2). EQ. 'B ')

のように FORTRAN 文法に従う任意の論理式 (1 行の FORTRAN ステートメントに収容できる範囲) を許すとする。

† A Simple Program Generation System by Macrocommand for Information Retrieval by TAKESHI KIYONO (Osaka Electro-Communication University), KATSUO IKEDA (Institute of Information Science and Electronics, The University of Tsukuba), and MASAOKI SHIMASAKI (Faculty of Engineering, Kyoto University).

†† 大阪電気通信大学

††† 筑波大学電子情報工学系学術情報処理センター

†††† 京都大学工学部情報工学教室

情報検索は検索条件式からなる論理関数サブプログラムを生成し、それが .TRUE. となるレコードのみ出力する形で行う。すなわち検索システムのソースプログラムのうち、検索条件論理関数サブプログラムのみ検索に応じて変化し、他の部分は固定である。固定部分はあらかじめ FORTRAN プログラムを作成し、コンパイルして目的プログラムを保存しておく。検索条件論理関数サブプログラムのソースプログラムはテキストエディタを用いて作成する。これについては次章に述べる。生成されたソースプログラムをコンパイルし、目的プログラム形式の主プログラムと結合して実行する。

3. 検索条件論理関数サブプログラムの生成

検索条件論理関数サブプログラム KEY はあらかじめそのプロトタイプを作成しておき、関数定義式の部分は

```
KEY=ZZZZZ
```

としておく。テキストエディタによりこの右辺をコマンドのパラメタの検索条件式と置き換える。これに必要なテキストエディタのコマンドはファイルとしてコマンドにより作成する。そのコマンドは次のとおりである。

```
CREATE filename, 80
$BUILD filename
RE
CH /ZZZZZ/@2/, 100
EN
$ENDB
```

CREATE, \$BUILD, \$ENDB はシステムコマンドで \$BUILD, \$ENDB の間のテキストを内容とする指定のファイルを作成する。@2 はマクロコマンドの第2のパラメタを指示し、この場合は検索条件式を与える。@n の部分はファイル作成時に文字列として実パラメタに置換される。RE, CH, EN はテキストエディタのコマンドである。RE は入力ファイルからエディットバッファへの読み込み、すなわちプロトタイプの検索条件論理関数サブプログラムの読み込みを指示する。CH /X/Y/, n は文字列 X を文字列 Y で置換することを示す。n は置換の対象の探索指定に関するパラメタである。EN はエディットの終了を指示する。これにより

```
KEY=検索条件論理式
```

となり、目的のソースプログラムが生成される。

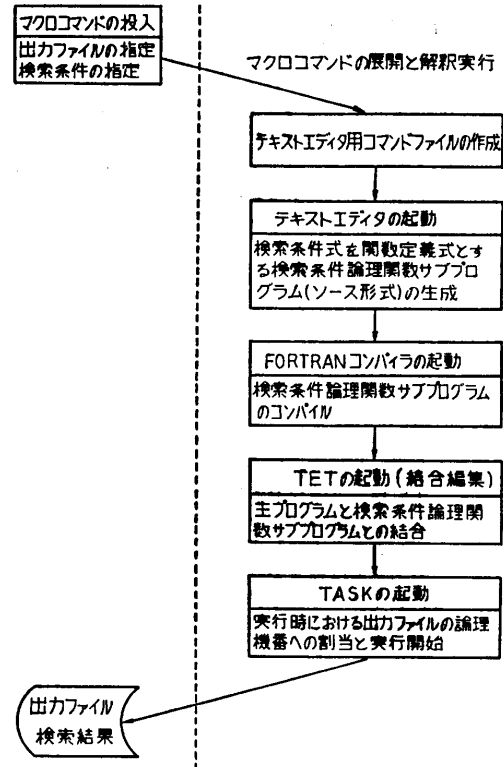


図 2 全体のジョブの流れ
Fig. 2 Schematic flow diagram of execution of macrocommand.

```
C ***** MAIN PROGRAM *****
LOGICAL KEY, KEYFUN
IMPLICIT INTEGER*2 (A-Z)
COMMON FIELD(256)
DATA NREC /600/
DO 10 I = 1, NREC
  READ(1) FIELD
C
C *****
C
C          KEYFUN = KEY(IDUMMY)
C          IF (KEYFUN) WRITE (2,200) I, (FIELD(K), K=81,90)
C *****
C
C 10 CONTINUE
C 200 FORMAT (I4,10A2)
C STOP
C END
C
C
C          LOGICAL FUNCTION KEY(IDUMMY)
C          IMPLICIT INTEGER*2 (A-Z)
C          COMMON FIELD(256)
C *****
C
C          KEY = ZZZZZ
C *****
C
C RETURN
C END
```

図 3 ソースプログラム
Fig. 3 Source program.

マクロコマンドの定義

```

BUILD SELECT.CSS
* SELECT OUT.FD, LOGICAL.EXPRESSION
*
*** BUILD COMMAND FILE FOR OS EDITOR
*
CREATE COMMAND.FIL, 80
$BUILD COMMAND.FIL
RE
CH /ZZZZZ/2/100
EN
$ENDB
*
*** EXECUTE OS EDITOR
*
CREATE KEY.SRC, 80
LO CSST, OSEEDIT
T CSST
AS 1, S010: SELECT.KEY, SR0
AS 2, KEY.SRC
AS 3, NULL:
AS 4, NULL:
AS 5, COMMAND.FIL
AS 6, NULL:
ST
DE COMMAND.FIL
*
*** COMPILE LOGICAL FUNCTION SUBPROGRAM
*
FORTCA KEY.SRC, KEY.OBJ
DE KEY.SRC
*
*** LINK EDIT SELECT.TSK
*
ESTA S010: SELECT.OBJ, SELECT.TSK, 128, , FORTRAN.RL, KEY.OBJ
DE KEY.OBJ
*
*** EXECUTE SELECT.TSK
*
LO CSST, SELECT.TSK
T CSST
AS 1, S010: MASTER.A00, SR0
AS 2, 01
ST
DE SELECT.TSK
*
*** END OF JOB
*
$EXIT
ENDB

```

投入マクロコマンド例

```
*SELECT TTY: , FIELD(91) . EQ. 'V ' . AND. FIELD(95) . EQ. 'A ' .
```

図 4 マクロコマンドの定義と投入マクロコマンドの実例

Fig. 4 Definition of macrommand and an example of the command used.

コマンドの投入から結果の出力までの過程を流れ図的に図 2 に示す。プログラムリストとマクロコマンドの定義をそれぞれ図 3、図 4 に示す。プログラムは方式の本質的な部分を明確にするためできるだけ簡潔にしてある。たとえば、EQUIVALECE 文により FIELD に適当な名前をつけることも可能で実用上は有用である。

4. おわりに

報告した情報検索方式の特徴はマクロコマンドによりソースプログラムを生成することである。検索の際にコンパイルされるのは検索条件に応じて変る論理関数サブプログラムのみであるので、ソースプログラム生成方式であるが、コンパイル時間は短い。検索条件として任意の論理式が指定できるので、かなりの汎用性があるといえる。簡易形であって、制限も強いが、作成に要する労力が小さいのが大きな特徴である。

使用した計算機は Interdata 7/32 で Command Substitution System¹⁾ と呼ばれるオペレータコマンド言語の機能が有用であった。コマンド言語とくに、そのマクロ機能は一般にもっと活用されるべきものと考えられる。

参 考 文 献

- 1) INTERDATA: OS/32 High Level Operator Command Substitution System (CSS) Package Users Manual, (1975).

(昭和 53 年 1 月 18 日受付)

(昭和 53 年 6 月 5 日採録)