

PL/ I による高速データ検索のために : REGIONAL(2) データセットの使用

大石, 範子
九州歯科大学内科

廣田, 安夫
九州歯科大学内科

<https://doi.org/10.15017/1474251>

出版情報 : 九州大学大型計算機センター広報. 12 (3), pp.239-245, 1979-09-20. 九州大学大型計算機センター
バージョン :
権利関係 :



PL/I による高速データ検索のために — REGIONAL (2) データセットの使用 —

大石 範子*, 廣田 安夫*

1. はじめに

ある地域一般住民の15年間にわたる長期追跡調査データを、データベースとして作成した。この多量のデータから、必要な情報を、即座にとり出し、高度な研究利用に役立てようとしているが、順編成のデータセットでは、必要なレコードを取り出す際には、データセットの先頭から順に、そのレコードに達するまで、アクセスしなければならない。その時間は、レコード数が多くなれば、無視できない。

そこで、我々は、他の編成法の1つである直接編成を試用した。直接編成のデータセットは、キーや区域番号をつかって1度のアクセスで、望むレコードを取り出せるという特徴をもつ。OSIV/F4 PL/Iでは、この編成をREGIONAL編成という。¹⁾ この編成法の使用により、データの検索に要する時間は、かなり縮小された。現在、まだ、センター配布の「利用の手引」には、この直接編成データセットの作成法などに関する情報が、記載されていないので、我々と同様に、多量データの検索を行なう方々の参考になればと思い、ここに概要を説明する。

2. 方 法

OSIV/F4 PL/IのREGIONAL編成には、REGIONAL(1)、REGIONAL(2)、REGIONAL(3)の3種の編成がある。¹⁾

REGIONAL編成ではデータは連続して番号づけられた区域に分割されるが、各区域はREGIONAL(1)、(2)編成では1個のレコードを、REGIONAL(3)編成では、1個以上のレコードを含む。REGIONAL(1)では区域番号を指定して直接アクセスする。REGIONAL(2)、(3)のレコードは、キーをもち、ユーザーは、転送文に区域番号とキーを共に指定してそのレコードを直接アクセスする。それぞれ長短があり、用途によって使い分けられる。¹⁾

ここでは、我々は、キーを用いて直接アクセスでき、1区域に1レコードを含むREGIONAL(2)を使用した。

2.1 REGIONAL(2)データセットの作成法

2.1.1 原始キーの指定

レコードは図1のような原始キーをもつ。¹⁾ 原始キーはレコードキー、区域番号から成る。これらは、レコードの直前に置かれる(ただし、埋めこまれたキーでなく、レコード長はキー長を含まない)。区域番号はレコードの位置をきめるために、レコードキーは、レコードを判別するために使われる。

*九州歯科大学内科

同じレコードキーをもつ複数個のレコードをかきこむと後のアクセスで、取り出せなくなるので、注意が必要である。¹⁾

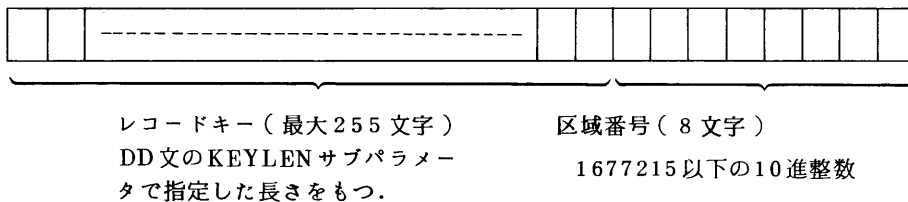


図 1.

転送文に指定される原始キーは、右側の8文字を区域番号として、左側の(DD文で指定されているキー長だけの)文字をレコードキーと解釈する。区域番号は1677215以下の10進整数でなければならない。先行する空白はゼロとみなされる。区域番号のなかに埋めこまれた空白があれば、最初の空白が区域番号を終了させる。原始キーは、指定されているキー長より短かければ、右側に、空白が付加される。原始キーの解釈の例を下に示す。

例題 2.1

KEY(▼120400000001▼)

DD文でKEYLEN = 4 と指定した場合

レコードキー 1204

区域番号 00000001

2.1.2 作成プログラム

OSIV/F4 PL/I では、直接編成のデータセットの作成法は次のようなものである。¹⁾

PL/I プログラムは、実行時ごとに異なるデータセットを処理できるように、ファイルと称するデータセットの記号表現を用いる。実行時に、ファイルを特定のデータセットと対応づけるのは、ジョブ制御言語の DD 文で指定する。ファイルは、種々のファイル属性を指定した DECLARE 文で、宣言する。

REGIONAL(2)データセットの作成は、DIRECT OUTPUT ファイルを用いて作成する。(SEQUENTIAL OUTPUT ファイルを用いても作成できる)¹⁾。ファイル作成時のポイントは、次の通りである。

① WRITE 文に区域番号とレコードキーを共に指定する。

② DD 文の指定

- i) UNIT 又は、VOLUME パラメータは、直接アクセス記憶装置を指定せねばならない。
- ii) DCB パラメータに KEYLEN オプションで、キー長を指定せねばならない。
- iii) RECFM = F でなければならない。

③ DECLARE 文

- i) FILE 属性は, RECORD OUTPUT DIRECT UNBUFFERED KEYED の属性が必要である.
- ii) ENVIRONMENT 属性に, REGIONAL(2)を指定する.

例題 2.2 KEYLEN = 3 の直接編成ファイル TEST を作成する.

○制御文

```
// EXEC PLIXCG, SYSOUT=R
//PLI.SYSIN DD *
```

ソ ー ス プ ロ グ ラ ム

```
//GO.SYSIN DD *
```

デ ー タ

```
//TEST DD DSN=F1234, TEST.DATA, DISP=(NEW,CATLG), UNIT=PUB,
//
//          SPACE=(TRK,10), DCB=(LRECL=60, RECFM=F,
//
//          DSORG=DA, KEYLEN=3)
//
```

○プログラム

** SOURCE STATEMENT LISTING **

```

1  TEST1 : PROC OPTIONS(MAIN) ;
2      DCL TEST FILE RECORD OUTPUT DIRECT UNBUFFERED KEYED
      ENV(REGIONAL(2)) ,
      INDATA          CHAR(60) ,
      ISKEYW          CHAR(6) ,
      VARKEY          CHAR(3) ,
      I               FIXED(3) ,
      ISKEY           CHAR(11) ;
3      OPEN FILE(TEST) ;
4      ON ENDFILE(SYSIN) GO TO OWARI ;
5      I=0 ;
6      RED1 : GET SKIP EDIT(INDATA)(A(60)) ;
7      I=I+1 ;
8      ISKEYW=I ;
9      VARKEY=SUBSTR(ISKEYW,4,3) ;
10     ISKEY=" " ;
11     SUBSTR(ISKEY,1,3)=VARKEY ;
12     SUBSTR(ISKEY,9,3)=VARKEY ;
13     WRITE FILE(TEST) FROM(INDATA) KEYFROM(ISKEY) ;
14     GO TO RED1 ;
15     OWARI : CLOSE FILE(TEST) ;
16     END TEST1 ;

```

注) SPACE は, 充分な領域を計算し指定する. また, 作成時には, 実レコードの他は, 指定領域のすべてにダミーレコードがかきこまれる関係で SYSTEM の EXCP 回数の制限値を越えるおそれが

ある (ERROR CODE = SE22) ので, JOB クラスを考慮した方がよい.

2.2 REGIONAL(2)データセットのアクセス法

ファイルアクセス時のポイントは次のようなものである.

- ① READ 文の KEY オプションにレコードキーを指定する.²⁾
- ② DD 文は, 2.1.2 において, SPACE パラメタを取り除き, DISP=OLD と変えること以外は 2.1.2 と同様.
- ③ DECLARE 文には, RECORD INPUT UNBUFFERED DIRECT KEYED の属性を指定. ENVIRONMENT 属性は, 2.1.2 と同じく REGIONAL(2) を指定.

例題 2.3 例題 2.2 で作成したファイルをアクセスする.

○制御文

```
// EXEC PLIXCG, SYSOUT=R
//PLI. SYSIN DD *
```

ソ ー ス プ ロ グ ラ ム

```
//GO. SYSIN DD *
//TEST DD DSN=F1234. TEST. DATA, DISP=OLD, UNIT=PUB,
//          DCB=( LRECL=60, RECFM=F, DSORG=DA, KEYLEN=3 )
//
```

○プログラム

** SOURCE STATEMENT LISTING **

STAT			
1	TEST2 : PROC OPTIONS(MAIN) ;		00000040
2	DCL TEST FILE RECORD INPUT DIRECT UNBUFFERED KEYED		00000050
	ENV(REGIONAL(2)) ,		00000060
	INDATA	CHAR(60) ,	00000070
	ISKEY	CHAR(3) ;	00000080
3	OPEN FILE(TEST) ;		00000100
4	ISKEY=" 3" ;		00000110
5	READ FILE(TEST) INTO(INDATA) KEY(ISKEY) ;		00000120
6	PUT EDIT("DATA=", INDATA)(SKIP,A,A) ;		00000130
7	CLOSE FILE(TEST) ;		00000140
8	END TEST2 ;		00000150

3. 使用例

実際に, 既に順編成のデータセットであったものにキーを加え, 直接編成のデータセットを作成し両方のアクセス時間の比較を行なった. 図2に, 使用したデータセットの構造とアクセスすべきレコードを示す. これは, S36年~S49年までのある町での検診結果の入ったデータセットであり, こ

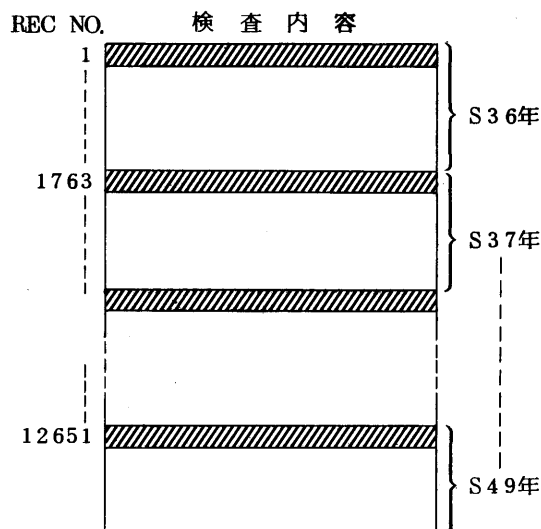


図2. CASE NO. = 1 の被検者の検査結果 [hatched box] をアクセスする

KEY IN CASE NO. : 64

CASE NO. = 64 *** NAME (MALE) *** -- DEAD --
DEATH DATE S.48.08.01.

DO YOU NEED AUTOPSY REPORT ? (YES OR NO) : NO

FOLLOW-UP ? (YES OR NO) : YES

DO YOU NEED FOLLOW UP ITEMS TABLE ? (YES OR NO) : NO

HOW MANY ITEMS DO YOU NEED ? : 2

KEY IN ITEM NO. : 4,5

START 17442151 >アクセスに要した時間22.01 秒
END17444352

FOLLOW-UP DATA CASE NO. = 64

* NAME (M) *

FOLLOW UP

ITEM	S36	S37	S38	S39	S40	S41	S42	S43	S456	S48	S49
DATE	4/28	.	4/25	.	4/27	5/16	6/08	5/	2/21	5/14	.
AGE	64	.	66	.	68	69	70	-	74	76	.
WEIGHT	52	.	52	.	55	52	49	52	50	49	.
SBP1	140	.	202	.	184	188	180	200	208	154	.

. NO ITEM
- NOT EXAMINED

図3. 順データセットをアクセスした場合

れを利用してある被検者の検診年ごとの結果を追跡しようとするものである。このアクセスを行う際に、順編成データセットでは、図3に示すように、約2.2秒のアクセス時間を必要としたのに対し、直接編成データセットでは、図4に示すように、約0.8秒を要したにすぎなかった。

```
START 17364277 > アクセスに要した時間 0.7 7 秒
END 17364354
FOLLOW-UP DATA CASE NO. = 64
```

```
*****
*               NAME               ( M )
*****

FOLLOW UP

ITEM      $36  $37  $38  $39  $40  $41  $42  $43  $456 $48  $49
DATE      4/28  . 4/25  . 4/27 5/16 6/08 5/   2/21 5/14  .
AGE        64   . 66   . 68   69   70   -   74   76   .
WEIGHT     52   . 52   . 55   52   49   52   50   49   .
  $BP1     140  . 202  . 184  188  180  200  208  154  .

.      NO ITEM
-      NOT EXAMINED
```

図4. 直接編成データセットをアクセスした場合

4. 考 察

3.で示したように、直接編成のデータセットを使用する事は、アクセス時間が短縮されるという長所がある。しかし、逆に、キーを含んだり、ランダムに記憶域に配置されることから、記憶領域が増大するという短所もある。同様に、順編成データセットと比較し、長所、短所をあげ表1に示す。

	長 所	短 所
直接編成 データセット	○ 1度のアクセスで欲しいレコードが、検索できる。 ○ ファイルの更新、挿入が、容易にできる。	○ キーの扱いなど、特異なもので扱いにくい。 ○ 外部記憶域が、大きくなる。
順編成 データセット	○ プログラム経験の少ない人にも扱いやすい。 ○ 外部記憶域が比較的少なくて済む。	○ 入力の際、欲しいレコードを検索するためにアクセス回数が、かさむ。 ○ ファイルの更新はできるが、挿入は困難である。

表1. PL/Iにおける直接編成データセットと順編成データセットの比較

5. おわりに

OSIV/F4 PL/Iの直接編成データセットREGIONAL(2)の作成法とアクセス法を試用し、一応の成果が得られた。筆者もまだ経験不十分であるが、これを機に、同様に多量のデータの検索を行なっている方々の参考になれば、幸いである。おわりに、有益な助言や御協力をいただいた九州大学大型計算機センターの鬼塚さん他、研究開発部の方々に深く感謝の意を表する。

参 考 文 献

- 1) 富士通：FACOM OSIV/F4 PL/I 使用手引書(64SP-3070-2) P61, P68~P71
- 2) 富士通：FACOM OSIV/F4 PL/I 文法書(64SP-3061-2) P151~P153