

ファイル比較プログラムFCMPの機能改善について

藤村, 直美
九州大学情報処理教育センター

<https://doi.org/10.15017/1468134>

出版情報：九州大学大型計算機センター広報. 18 (6), pp.601-611, 1985-11-25. 九州大学大型計算機センター
バージョン：
権利関係：

ファイル比較プログラムFCMPの機能改善について

藤 村 直 美*

1. はじめに

TSSの普及につれてテキストエディタを利用してソーステキストやデータを作成・編集する機会が増えている。それに伴って、例えば、テキストエディタによって意図しない修正が行われたにもかかわらずそれに気が付かないとか、同じようなテキストファイルが沢山できて、それらの異同が正確にはわからないといったような種々の問題が発生する。

こうした状況を改善するために、二つの類似のテキストファイルを比較して、挿入・削除・修正といった関係を行単位で表示するソフトウェアツール FCMP (File CoMPare) を作成し、九州大学大型計算機センターにおいても昭和54年以来提供してきた[1]。その後FCMPについて利用者から寄せられたいくつかの要望に基づいて、今回次のような機能改善を行った。

- ① 旧FCMPではレコード長80バイトの固定長レコード形式(FB)のデータセットしか比較できなかったが、新FCMPでは固定長または可変長レコード形式(VB)で、任意のレコード長のデータセットを比較できる。ただし実際に比較するのは行番号などを除いて各レコードの先頭から220バイトまでである。
- ② 新FCMPでは指定したデータセット名を結果に含めて表示する。旧FCMPでは比較するデータセット名が結果に含まれていなかったため、もともとよく似たデータセットを比較することもあり、プリンタに出力した結果を後から確認するときなど、どちらがどちらかわからなくなり混乱することがあった。
- ③ 旧FCMPでは比較結果の前後に表示される同一の'まとめ'をすべて同じ出力先へ表示していたが、新FCMPでは最初の'まとめ'は必ずTSS端末へ表示し、比較結果と最後の'まとめ'を一緒に指定された出力先へ表示する。これによって比較結果をプリンタへ出力したときでもTSS

昭和60年9月9日受理

*九州大学情報処理教育センター

S 端末で結果の概略を把握することができる。

- ④ 旧FCMPでは行番号の位置に無関係にレコードの最初の72バイトを比較の対象としていたが、新FCMPでは行番号を除いて最高220バイトまでのテキストを比較する。そのため例えばCOBOLプログラムで行を挿入・削除して行番号をつけ直しても、特別な配慮をせずに意味のある比較を行うことができる。
- ⑤ 比較の結果は各レコードの内容に、データセットの先頭からの行数ではなく、実際にデータセットの各レコードに付加されている行番号をつけて表示する。

2. ファイル比較プログラム

図1(a)に示すような6行からなるテキストファイルfile1を考える。これを修正して図1(b)のfile2が得られたものとしよう。すなわちfile1の2行目をYELLOWからGREENに変更、3行目と4行目の間にWHITEを挿入したものである。こうしてできたfile2をさらに続けて更新すると同じようなファイルが沢山でき上がってしまう。この例のように、それぞれのファイルの行数が少ないときは人間の目で確かめることも可能である。しかし、数百行あるいは数千行からなる、意識的に変更した所だけ異なるとは限らない二つのファイルの異同を人間の目で確認することは不可能といってよいだろう。こうした仕事は計算機に任せる方が正確で速い。

00010 RED	00010 RED
00020 YELLOW	00020 GREEN
00030 BLUE	00030 BLUE
00040 RED	00031 WHITE
00050 BLACK	00040 RED
00060 YELLOW	00050 BLACK
	00060 YELLOW

(a) file1 (b) file2

図1 ファイル修正例

ファイル比較プログラムFCMPは入力部、比較部、出力部からなる。入力部は比較処理に備えて

対象となるテキストファイルを行単位でコード化する。比較部はその情報をもとにして、'一意に等しい行'、'等しい行'、'等しくない行'、'挿入された行'、'削除された行'の5種類に分類する。出力部は利用者が指定したパラメータにしたがって、比較の結果を編集して出力する（4種類の出力形式が可能）。

前述の図1において、'一意に等しい行'とはfile1とfile2に一行ずつあるBLUEとBLACKの各行である。これらの行はそれぞれのファイルにただ一行しかなく、しかも互いに等しい。REDはfile1とfile2に各2行あり、YELLOWはfile1に2行、file2に1行あるので'等しい行'である。file2にはGREENとWHITEがあるが、file1にはない。行の関係を表示するために、まず最初に一意に等しい行の対応を、次に等しい行の対応をとって並べてみると図2のようになる。ここで中央の==は左右の行が一意に等しいこと、空白は左右の行が等しいが、一意には等しくないこと、**は左右の行が等しくないこと、>はfile2へ行が挿入されていることを示している。

1	RED		RED	等しい行
2	YELLOW	**	GREEN	等しくない行
3	BLUE	==	BLUE	一意に等しい行
4		>	WHITE	挿入された行
5	RED		RED	等しい行
6	BLACK	==	BLACK	一意に等しい行
7	YELLOW		YELLOW	等しい行

図2 file1とfile2の異同

FCMPはパラメータの組合せによって4種類の出力形式を用意している。以下にパラメータの組合せによる出力例を中心に結果の見方について述べる。

パラメータは、

D:全ての行を出力する (Detail)

S:一意に等しい行と等しい行とは出力しない (Simple)

N:幅の狭い出力装置向 (Narrow)

W: 幅の広い出力装置向 (Wide)

の組合せである。以下の四つの例で比較の対象となる2つのファイルはすべて同じものを用いている。

図3にパラメータDWによる出力例を示す。左側のファイルを訂正・編集して右側のファイルが得られたものとする。図の中央の==, 空白, **, >は図2の説明で述べたのと同じ意味である。<はその行が右のファイルでは削除されたこと（左のファイルへ挿入されたこと）を示している。'. 'はその行が1行（55字）に入りきらないので複数行に折り曲げて表示されていることを示している。

さらに具体的に説明する。左の行番号10の行（以後行10というように略す）と右の行10は一意に等しい。左の行20と右の行20は等しいけれども、左の行20は左の行50, 120, ...に同一の行があり、右の行20は右の行40, 行70, ...に同一の行がある。従って一意に等しくはない。左の行30と右の行30は等しくない。右のレコードが長くて1行に入らないので次の行に継続して表示されている。右の行40と行50は挿入されている。FCMPは行単位で等しいか否かを判断するので、左の行30と右の行50がよく似ているというような対応付けは行わない。左の行130から行160の内容をまとめて右の行150にしたため、左の行130と右の行150は等しくなく、左の行140から行160は削除されている。

パラメータDWによる出力は2つのファイルが左右に分離され、全行出力されるので、ソースリストとしても十分役にたつ。ところで、この例ではソーステキストが約40行と少いので全行出力しても負担にならないが、大きなテキストファイルでは全部の情報を出力するのは不経済であり、異なる部分だけわかった方がむしろ都合がよいことの方が多い。そこで、一意に等しい行および等しい行を出力しないようにしたのが図4に示すパラメータSWによる出力例である。中央の記号の意味は同じであるが、一意に等しい行および等しい行は出力されていない。この2種類の行が何行あったかは出力リストの最後に示される'まとめ'を見ればわかる。Eは等しい行（10行）、=は一意に等しい行（23行）である。

```

DSN1 TEXT                                DSN2 TEXT
00000010 # THIS PROGRAM COMPUTES N PRIME NUMBERS. == 00000010 # THIS PROGRAM COMPUTES N PRIME NUMBERS.
00000020 #                                00000020 #
00000030 INTEGER PRIME(1000)                ** 00000030 DEFINE(LIMIT,1000) # SIZE OF TABL
                                           .. E
                                           > 00000040 #
                                           > 00000050 INTEGER PRIME(LIMIT)
                                           == 00000060 LOGICAL SW
                                           00000070 #
00000040 LOGICAL SW                        == 00000080 READ(5,10) N; 10 FORMAT(I5)
00000050 #                                ** 00000090 IF ( N < 1 I N > LIMIT) # INPUT DATA C
00000060 READ(5,10) N; 10 FORMAT(I5)
00000070 IF ( N < 1 & N > 1000)
                                           .. HECK
00000080 *(                                00000100 *(
00000090 WRITE(6, 20) N; 20 FORMAT( ' INVALID N (', I ** 00000110 WRITE(6,20) N; 20 FORMAT(5X,' INVALID N (',I
5, ')') )                                .. 6,')') )
00000100 STOP                                == 00000120 STOP
00000110 *(                                00000130 *(
00000120 #                                00000140 #
00000130 PRIME(1) = 2                        ** 00000150 PRIME(1) = 2; NO = 1; INDEX = 1; NUM = 1;
00000140 NO = 1                                <
00000150 INDEX = 1                            <
00000160 NUM = 1                                <
00000170 #                                00000160 #
00000180 REPEAT                                == 00000170 REPEAT
00000190 *(                                00000180 *(
00000200 NUM = NUM + 2                        == 00000190 NUM = NUM + 2
00000210 IF (PRIME(INDEX) ** 2 <= NUM) INDEX = INDEX + 1 == 00000200 IF (PRIME(INDEX) ** 2 <= NUM) INDEX = INDEX + 1
00000220 SW = .TRUE.                        == 00000210 SW = .TRUE.
00000230 FOR (I = 1; I <= INDEX; I = I + 1) == 00000220 FOR (I = 1; I <= INDEX; I = I + 1)
00000240 IF (MOD(NUM, PRIME(I)) == 0)        == 00000230 IF (MOD(NUM, PRIME(I)) == 0)
00000250 *(                                == 00000240 *(
00000260 SW = .FALSE.                        == 00000250 SW = .FALSE.
00000270 BREAK                                == 00000260 BREAK
00000280 *(                                == 00000270 *(
00000290 IF (SW)                            == 00000280 IF (SW)
00000300 *(                                == 00000290 *(
00000310 NO = NO + 1                        == 00000300 NO = NO + 1
00000320 PRIME(NO) = NUM                    == 00000310 PRIME(NO) = NUM
00000330 *(                                == 00000320 *(
00000340 *(                                00000330 *(
00000350 UNTIL (NO >= N)                    == 00000340 UNTIL (NO >= N)
00000360 #                                == 00000350 #
00000370 WRITE(6,30) (PRIME(I),I=1,N); 30 FORMAT(5X,10I10 == 00000360 WRITE(6,30) (PRIME(I),I=1,N); 30 FORMAT(5X,10I10
)                                           .. )
00000380 STOP                                == 00000370 STOP
00000390 #                                00000380 #
00000400 END                                == 00000390 END

*** FCMP SUMMARY REPORT ***

DSN1 = 40 LINES. (EXAMPLE.TEXT(LEFT)      )
DSN2 = 39 LINES. (EXAMPLE.TEXT(RIGHT)     )

E: 10; == 23; *: 4; >: 2; <: 3;

FCMP NORMAL END

```

図3 パラメータDWによる出力例

```
FILE COMPARE PROGRAM V-02 L-01 -850807- DATE 10/21/85 TIME 14:27:35

DSN1 TEXT DSN2 TEXT # SIZE OF TABL
00000030 INTEGER PRIME(1000) ** 00000030 E ,
    .> 00000040 # ,
    > 00000050 INTEGER PRIME(LIMIT)
    ** 00000090 IF ( N < 1 I N > LIMIT) # INPUT DATA C
    HECK WRITE(6,20) N; 20 FORMAT(5X,' INVALID N (',I
    6,')' ) PRIME(1) = 2; NO = 1; INDEX = 1; NUM = 1;
    ** 00000150 PRIME(1) = 2; NO = 1; INDEX = 1; NUM = 1;
    <
    <
    <

*** FCMP SUMMARY REPORT ***
DSN1 = 40 LINES. (EXAMPLE.TEXT(LEFT)
DSN2 = 39 LINES. (EXAMPLE.TEXT(RIGHT)
E: 10; =: 23; *: 4; >: 2; <: 3;

FCMP NORMAL END
```

図4 パラメータSWによる出力例

パラメータ W を指定したときの出力形式は幅が 80 欄の T S S 端末には不向きである。そこで、80 欄に納まるようにしたのが、図 5 に示したパラメータ DN による出力例である。左側に表示されている 2 列の行番号の間の記号が図 3 で述べた記号と同じ意味を持っている。異なる行は左のファイル

```

FILE COMPARE PROGRAM V-02 L-01 -850807- DATE 10/21/85 TIME 14:28:02

DSN1      DSN2      TEXT
00000010=00000010 #      THIS PROGRAM COMPUTES N PRIME NUMBERS.
00000020 00000020 #
00000030*      INTEGER PRIME(1000)
      *00000030      DEFINE(LIMIT,1000) # SIZE OF TABL
      -      E
      >00000040 #
      >00000050      INTEGER PRIME(LIMIT)
00000040=00000060      LOGICAL SW
00000050 00000070 #
00000060=00000080      READ(5,10) N; 10 FORMAT(15)
00000070*      IF ( N < 1 & N > 1000)
      *00000090      IF ( N < 1 I N > LIMIT) # INPUT DATA C
      -      HECK
00000080 00000100      ¥(
00000090*      WRITE(6,20) N; 20 FORMAT( ' INVALID N (', I
      -      5, ' )' )
      *00000110      WRITE(6,20) N; 20 FORMAT(5X,' INVALID N (',I
      -      6, ' )' )
00000100=00000120      STOP
00000110 00000130      ¥)
00000120 00000140 #
00000130*      PRIME(1) = 2
      *00000150      PRIME(1) = 2; NO = 1; INDEX = 1; NUM = 1;
00000140<      NO = 1
00000150<      INDEX = 1
00000160<      NUM = 1
00000170 00000160 #
00000180=00000170      REPEAT
00000190 00000180      ¥(
00000200=00000190      NUM = NUM + 2
00000210=00000200      IF (PRIME(INDEX) ** 2 <= NUM) INDEX =INDEX + 1
00000220=00000210      SW = .TRUE.
00000230=00000220      FOR (I = 1; I <= INDEX; I = I + 1)
00000240=00000230      IF (MOD(NUM, PRIME(I)) == 0)
00000250=00000240      ¥(
00000260=00000250      SW = .FALSE.
00000270=00000260      BREAK
00000280=00000270      ¥)
00000290=00000280      IF (SW)
00000300=00000290      ¥(
00000310=00000300      NO = NO + 1
00000320=00000310      PRIME(NO) = NUM
00000330=00000320      ¥)
00000340 00000330      ¥)
00000350=00000340      UNTIL (NO >= N)
00000360 00000350 #
00000370=00000360      WRITE(6,30) (PRIME(I),I=1,N); 30 FORMAT(5X,10I10
      -      )
00000380=00000370      STOP
00000390 00000380 #
00000400=00000390      END

*** FCMP SUMMARY REPORT ***

DSN1 = 40 LINES. (EXAMPLE.TEXT(LEFT) )
DSN2 = 39 LINES. (EXAMPLE.TEXT(RIGHT) )

E: 10; := 23; *: 4; >: 2; <: 3;
FCMP NORMAL END

```

図 5 パラメータ DN による出力例

を1行出力した後に右のファイルを1行出力する。図5ではテキストの違いのうちでも特に欄の移動や文字の置き換えがわかり易い。例えば、左の行70と右の行90は'&'と'|'が一文字置き換わっていること、右の行90に註釈が追加されていることがわかる。また、左の行90と右の行110では空白の挿入・削除や文字の置き換えが行われていることがよくわかる。パラメータDNでは出力量が多過ぎることが多いので、図6に示すようにパラメータSNを用いて異なる部分のみに出力を限定することができる。

```

FILE COMPARE PROGRAM V-02 L-01 -850807- DATE 10/21/85 TIME 14:27:05

DSN1      DSN2      TEXT
00000030*          INTEGER PRIME(1000)
          *00000030      DEFINE(LIMIT,1000)                # SIZE OF TABL
          .              E
          >00000040      #
          >00000050          INTEGER PRIME(LIMIT)
00000070*          IF ( N < 1 & N > 1000)
          *00000090      IF ( N < 1 | N > LIMIT)                # INPUT DATA C
          .              HECK
00000090*          WRITE(6, 20) N; 20 FORMAT( ' INVALID N (', I
          .              5, ' ) ' )
          *00000110      WRITE(6,20) N; 20 FORMAT(5X,' INVALID N (',I
          .              6, ' ) ' )
00000130*          PRIME(1) = 2
          *00000150      PRIME(1) = 2; NO = 1; INDEX = 1; NUM = 1;
00000140<          NO = 1
00000150<          INDEX = 1
00000160<          NUM = 1

*** FCMP SUMMARY REPORT ***

DSN1 = 40 LINES. (EXAMPLE.TEXT(LEFT) )
DSN2 = 39 LINES. (EXAMPLE.TEXT(RIGHT) )

E: 10; =: 23; *: 4; >: 2; <: 3;

FCMP NORMAL END

```

図6 パラメータSNによる出力例

3. 使用方法

3.1 FCMP コマンド

(1) 機能

よく似た二つのテキストファイルと比較し、挿入・削除・修正などの関係をレコード単位で表示する。

(2) コマンド形式

コマンド名	オペランド
FCMP	DSN1 DSN2 [PARM (<u>SN</u> SW DN DW)] [SY (<u>*</u> 出力クラス)]

オペランドの説明

DSN1 : 比較したいテキストファイルを指定する。比較結果の左のファイルとなる。

DSN2 : 比較したいテキストファイルを指定する。比較結果の右のファイルとなる。

PARM : 比較結果の出力形式を指定する。パラメータの組合わせとしては、SN, SW, DN, DWの4つがある。パラメータの意味については第2節を参照されたい。特に指定しないとSN（最も簡潔な出力形式）を指定したものとみなす。

SY : 結果の出力先を指定する。特に指定しないと '*' を指定したものとみなし、TSS 端末に結果を表示する。日本語ラインプリンタなどに結果を出力したいときは、出力クラス (O, K, S, U など) を指定する。

3.2 使用例

- ① 最も簡単な形式（パラメータ SN）で T S S 端末へ結果を表示する。

```

READY
FCMP EXAMPLE.TEXT(LEFT) EXAMPLE.TEXT(RIGHT)
*** FCMP SUMMARY REPORT ***

DSN1 = 40 LINES. (EXAMPLE.TEXT(LEFT) )
DSN2 = 39 LINES. (EXAMPLE.TEXT(RIGHT) )

E: 10; =: 23; *: 4; >: 2; <: 3;

FILE COMPARE PROGRAM V-02 L-01 -850807- DATE 09/04/85 TIME 09:14:18

DSN1      DSN2      TEXT
00000030*          INTEGER PRIME(1000)
          *00000030      DEFINE(LIMIT,1000)          # SIZE OF TABL
          -              E
          >00000040      #
          >00000050          INTEGER PRIME(LIMIT)
00000070*          IF ( N < 1 & N > 1000)
          *00000090      IF ( N < 1 I N > LIMIT)          # INPUT DATA C
          -              HECK
00000090*          WRITE(6, 20) N; 20 FORMAT( ' INVALID N (' , I
          -              5, ' )' )
          *00000110      WRITE(6,20) N; 20 FORMAT(5X,' INVALID N (' ,I
          -              6, ' )' )
00000130*          PRIME(1) = 2
          *00000150      PRIME(1) = 2; NO = 1; INDEX = 1; NUM = 1;
00000140<          NO = 1
00000150<          INDEX = 1
00000160<          NUM = 1

*** FCMP SUMMARY REPORT ***

DSN1 = 40 LINES. (EXAMPLE.TEXT(LEFT) )
DSN2 = 39 LINES. (EXAMPLE.TEXT(RIGHT) )

E: 10; =: 23; *: 4; >: 2; <: 3;

FCMP NORMAL END
KQC0550I ENTRY (A) F1234.0ZYXL DELETED
KQC0550I ENTRY (A) F1234.0ZYXR DELETED
READY

```

- ② 詳細な出力形式（パラメータ DW）で日本語ラインプリンタ（出力クラス U）へ結果を出力する。

```

READY
FCMP EXAMPLE.TEXT(LEFT) EXAMPLE.TEXT(RIGHT) PARM(DW) SY(U)
*** FCMP SUMMARY REPORT ***

DSN1 = 40 LINES. (EXAMPLE.TEXT(LEFT) )
DSN2 = 39 LINES. (EXAMPLE.TEXT(RIGHT) )

E: 10; =: 23; *: 4; >: 2; <: 3;

KQC0550I ENTRY (A) F1234.0ZYXL DELETED
KQC0550I ENTRY (A) F1234.0ZYXR DELETED
READY

```

3.3 使用上の注意

- ① 比較の対象となるファイルのレコード数は各々 9 9 9 8 以下でなければならない。
- ② 比較の対象となるファイルのレコード形式は、固定長 (FB) と可変長 (VB) のどちらでもよい。
- ③ 各レコードの行番号を除いた 2 2 0 バイトまでが比較の対象となる。
- ④ テキストが 1 行 (5 5 字) に入り切らないときは、次の行へ続けて表示される。その場合は比較の結果を示す欄に ' . ' が付加される。
- ⑤ 幅の広い T S S 端末にパラメータ W を指定して出力するときは、TERMINAL コマンドで LINE SIZE を 1 3 3 以上 (左端の制御文字 1 字を含む) に変更する必要がある。
- ⑥ FCMP は本来よく似たテキストファイルを比較することを前提としているので、全く異なる二つのファイルを比較すると、CPU 時間を極端に多く必要とする。

参考文献

- [1] 藤村 : テキストファイルを比較するソフトウェアツール FCMP の使用について, 九州大学大型計算機センター広報, Vol. 12, No. 4, pp. 367-375, 1979