

## 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

藤村, 直美  
九州大学工学部情報工学科

<https://doi.org/10.15017/1474870>

---

出版情報 : 九州大学大型計算機センター広報. 12 (4), pp.367-375, 1979-12-15. 九州大学大型計算機センター  
バージョン :  
権利関係 :

## 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

藤 村 直 美\*

## 1. はじめに

TSS の普及につれてオンライン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、従来は経験しなかったような種々の問題が発生してきた。例えば、オンライン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、それらの相互間の異同を確認する必要が生ずるといったようなことである。

図1(a) に示すような6行からなるテキストファイル file 1 を考える。これを修正して図1(b) の file 2 が得られたものとしよう。すなわち file 1 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 file 2 をさらに更新を繰り返すと同じようなファイルが沢山できてしまう。この例のように、それぞれのファイルの行数が少い時は人間の目で確かめることも可能である。しかし、100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) を作成した。このような基本的なソフトウェアツールはソフトウェアの開発・改善・保守等に役立ち、その応用範囲も広い。以下、FCMP の使用方法を中心に述べる。

1	RED	1	RED
2	YELLOW	2	GREEN
3	BLUE	3	BLUE
4	RED	4	WHITE
5	BLACK	5	RED
6	YELLOW	6	BLACK
		7	YELLOW

(a) file 1                      (b) file 2

図1 ファイル修正例

1	RED		RED
2	YELLOW	**	GREEN
3	BLUE	==	BLUE
4		>	WHITE
5	RED		RED
6	BLACK	==	BLACK
7	YELLOW		YELLOW

図2 file 1 と file 2 の異同

## 2. ファイル比較プログラム

FCMP は最初九州大学工学部情報工学科のFACOM230-45S OSIIにおいて作成され、後に九州大学大型計算機センターのFACOM M-190 OSIV/F4に移し換えられた。手続きは記述の容

\* 九州大学工学部情報工学科

易さ、処理速度、プログラムの大きさ、移し換えの容易さ等を考慮して、COBOLで記述している。

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▼行の組、▼等しい▼行の組、▼等しくない▼行の組、▼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従って、必要な情報を編集して出力する(4種類の組合せが可)。

前述の図1において、▼一意に等しい▼行の組とはfile 1とfile 2に一行ずつあるBLUEとBLACKの各行である。REDはfile 1とfile 2に各2行あり、YELLOWはfile 1に2行、file 2に1行あるので▼等しい▼行の組である。file 2にはGREENとWHITEがあるが、file 1にはない。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file 2へ行が挿入されていることを示している。

前述したように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。以下の例で比較の対象となる2つのファイルは4例とも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。

さらに具体的に説明する。左の1行目と右の1行目は一意に等しい。左の2行目と右の2行目は等しいけれども、左の2行目は左の5行目、12行目、…に同一の行があり、右の2行目は右の4行目、7行目、…に同一の行がある。従って一意に等しくはない。左の3行目と右の3行目は等しくない。右の4行目と5行目は挿入されている。FCMPは行単位で等しいか否かを判断するので、左の3行目と右の5行目がよく似ているというような対応付けは行わない。左の13行目から16行目の内容をまとめて右の15行目にしたため、左の13行目と右の15行目は等しくなく、左の14行目から16行目は削除されている。

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立つ(制限を参照)。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、異なる部分だけが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の組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、==は一意に等しい行である。

FCMP EXAMPLE.TEXT(LEFT) EXAMPLE.TEXT(RIGHT) PARM(DW)

```

FACOM OSIV/F4 FCMP -790811- DATE 08/20/79 TIME 10:35:02 (N.FUJIMURA)
DSN1 TEXT DSN2 TEXT
0001 # THIS PROGRAM COMPUTES N PRIME NUMBERS. == 0001 # THIS PROGRAM COMPUTES N PRIME NUMBERS.
0002 # == 0002 #
0003 # INTEGER PRIME(1000) ** 0003 # DEFINE(LIMIT,1000) # SIZE OF TABLE
> 0004 #
> 0005 # INTEGER PRIME(LIMIT)
== 0006 # LOGICAL SW
0004 # LOGICAL SW == 0007 #
0005 # == 0008 # READ(5,10) N; 10 FORMAT(I5)
0006 # READ(5,10) N; 10 FORMAT(I5) == 0009 # IF ( N < 1 & N > 1000) ** 0009 # IF ( N < 1 | N > LIMIT) # INPUT DATA CHECK
0007 # IF ( N < 1 & N > 1000) $ (
0008 # $ ( ** 0010 # $ (
0009 # WRITE(6, 20) N; 20 FORMAT( ' INVALID N (', I5, ' ' ** 0011 # WRITE(6,20) N; 20 FORMAT(5X,' INVALID N (' ,I6,')
0010 # STOP == 0012 # STOP
0011 # $) 0013 # $)
0012 # 0014 #
0013 # PRIME(1) = 2 ** 0015 # PRIME(1) = 2; NO = 1; INDEX = 1; NUM = 1;
0014 # NO = 1 <
0015 # INDEX = 1 <
0016 # NUM = 1 <
0017 # 0016 #
0018 # REPEAT == 0017 # REPEAT
0019 # $ ( == 0018 # $ (
0020 # NUM = NUM + 2 == 0019 # NUM = NUM + 2
0021 # IF (PRIME(INDEX) ** 2 <= NUM) INDEX =INDEX + 1 == 0020 # IF (PRIME(INDEX) ** 2 <= NUM) INDEX =INDEX + 1
0022 # SW = .TRUE. == 0021 # SW = .TRUE.
0023 # FOR (I = 1; I <= INDEX; I = I + 1) == 0022 # FOR (I = 1; I <= INDEX; I = I + 1)
0024 # IF (MOD(NUM, PRIME(I)) == 0) == 0023 # IF (MOD(NUM, PRIME(I)) == 0)
0025 # $ ( == 0024 # $ (
0026 # SW = .FALSE. == 0025 # SW = .FALSE.
0027 # BREAK == 0026 # BREAK
0028 # $) == 0027 # $)
0029 # IF (SW) == 0028 # IF (SW)
0030 # $ ( == 0029 # $ (
0031 # NO = NO + 1 == 0030 # NO = NO + 1
0032 # PRIME(NO) = NUM == 0031 # PRIME(NO) = NUM
0033 # $) == 0032 # $)
0034 # 0033 # $)
0035 # UNTIL (NO >= N) == 0034 # UNTIL (NO >= N)
0036 # 0035 #
0037 # WRITE(6,30) (PRIME(I),I=1,N); 30 FORMAT(5X,10I10) == 0036 # WRITE(6,30) (PRIME(I),I=1,N); 30 FORMAT(5X,10I10)
0038 # STOP == 0037 # STOP
0039 # 0038 #
0040 # END == 0039 # END
DSN1 = 40 LINES DSN2 = 39 LINES
E: 10; =: 23; *: 4; >: 2; <: 3;
FCMP NORMAL END
READY

```

図3. パラメータ DW による出力例

```

FCMP EXAMPLE.TEXT(LEFT) EXAMPLE.TEXT(RIGHT) PARM(SW)
FACOM OSIV/P4 FCMP -790811- DATE 08/20/79 TIME 10:31:51 (N.FUJIMURA)
DSN1 TEXT DSN2 TEXT
0003 INTEGER PRIME(1000) ** 0003 DEFINE(LIMIT,1000) # SIZE OF TABLE
> 0004 #
> 0005 INTEGER PRIME(LIMIT)
0007 IF ( N < 1 & N > 1000) ** 0009 IF ( N < 1 | N > LIMIT) # INPUT DATA CHECK
0009 WRITE(6,20) N; 20 FORMAT( ' INVALID N (', I5, ' ' ** 0011 WRITE(6,20) N; 20 FORMAT(5X,' INVALID N (',I6,',')
0013 PRIME(1) = 2 ** 0015 PRIME(1) = 2; NO = 1; INDEX = 1; NUM = 1;
0014 NO = 1
0015 INDEX = 1
0016 NUM = 1
DSN1 = 40 LINES DSN2 = 39 LINES
E: 10; #: 23; *: 4; >: 2; <: 3;
FCMP NORMAL END
READY

```

図4. パラメータ SW による出力例

パラメータ W を指定した出力は幅が 80 欄の TSS 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。左端の行番号間の記号が図 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を 1 行出力した後右のファイルを 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 7 行目と右の 9 行目は & と | が一文字置き換わっていること、右の 9 行目に注釈が追加されていることがわかる。また、左の 9 行目と右の 11 行目では空白の挿入・削除・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 を用いて出力を異なる部分のみに限定することができる。

### 3. おわりに

FCMP を利用すると(1)テキスト編集、(2)デバッグ、(3)テキストファイルの保守等を能率良く行うことができる。また、その他の応用範囲も広い。こうした基本的なソフトウェアツールは便利で役に立つにもかかわらず、一般の関心は高くないようである。このような道具を存分に利用して、より快適な環境で仕事をする事が望まれる。

### 参考文献

1. Heckel, P. A technique for isolating differences between files. CACM, Vol. 21, No. 4, pp. 264-268, 1978.
2. 角田, 前野, 佐渡 ファイル比較技法を利用したソフトウェアツールについて, 情報処理学会第 20 回全国大会, pp. 359-360, 1979.
3. 藤村, 牛島 プログラムの移換性からみた FORTRAN と COBOL の比較, 九大工学集報, Vol. 52, No. 2, pp. 157-162, 1979.

```

FCMP EXAMPLE.TEXT(LEFT) EXAMPLE.TEXT(RIGHT) PARM(DN)
FACOM OSIV/F4 FCMP -790811- DATE 08/20/79 TIME 10:33:10 (N.FUJIMURA)
DSN1 DSN2 TEXT
0001=0001 # THIS PROGRAM COMPUTES N PRIME NUMBERS.
0002 0002 #
0003* INTEGER PRIME(1000)
*0003 DEFINE(LIMIT,1000) # SIZE OF TABLE
>0004 #
>0005 INTEGER PRIME(LIMIT)
0004=0006 LOGICAL SW
0005 0007 #
0006=0008 READ(5,10) N; 10 FORMAT(I5)
0007* IF ( N < 1 & N > 1000)
*0009 IF ( N < 1 | N > LIMIT) # INPUT DATA CHECK
0008 0010 $(
0009* WRITE(6, 20) N; 20 FORMAT(' INVALID N (' , I5, ') ' )
*0011 WRITE(6,20) N; 20 FORMAT(5X,' INVALID N (' ,I6,') ' )
0010=0012 STOP
0011 0013 $(
0012 0014 #
0013* PRIME(1) = 2
*0015 PRIME(1) = 2; NO = 1; INDEX = 1; NUM = 1;
0014< NO = 1
0015< INDEX = 1
0016< NUM = 1
0017 0016 #
0018=0017 REPEAT
0019 0018 $(
0020=0019 NUM = NUM + 2
0021=0020 IF (PRIME(INDEX) ** 2 <= NUM) INDEX =INDEX + 1
0022=0021 SW = .TRUE.
0023=0022 FOR (I = 1; I <= INDEX; I = I + 1)
0024=0023 IF (MOD(NUM, PRIME(I)) == 0)
0025=0024 $(
0026=0025 SW = .FALSE.
0027=0026 BREAK
0028=0027 $(
0029=0028 IF (SW)
0030=0029 $(
0031=0030 NO = NO + 1
0032=0031 PRIME(NO) = NUM
0033=0032 $(
0034 0033 $(
0035=0034 UNTIL (NO >= N)
0036 0035 #
0037=0036 WRITE(6,30) (PRIME(I),I=1,N); 30 FORMAT(5X,10I10)
0038=0037 STOP
0039 0038 #
0040=0039 END
DSN1 = 40 LINES DSN2 = 39 LINES
E: 10; =: 23; *: 4; >: 2; <: 3;
FCMP NORMAL END
READY

```

図5. パラメータ DN による出力例

```

FCMP EXAMPLE.TEXT(LEFT) EXAMPLE.TEXT(RIGHT) PARM(SN)
FACOM OSIV/F4 FCMP -790811- DATE 08/20/79 TIME 10:28:53 (N.FUJIMURA)
DSN1 DSN2 TEXT
0003* INTEGER PRIME(1000)
*0003 DEFINE(LIMIT,1000) # SIZE OF TABLE
>0004 #
>0005 INTEGER PRIME(LIMIT)
0007* IF ( N < 1 & N > 1000)
*0009 IF ( N < 1 | N > LIMIT) # INPUT DATA CHECK
0009* WRITE(6, 20) N; 20 FORMAT(' INVALID N (' , I5, ') ' )
*0011 WRITE(6,20) N; 20 FORMAT(5X,' INVALID N (' ,I6,') ' )
0013* PRIME(1) = 2
*0015 PRIME(1) = 2; NO = 1; INDEX = 1; NUM = 1;
0014< NO = 1
0015< INDEX = 1
0016< NUM = 1
DSN1 = 40 LINES DSN2 = 39 LINES
E: 10; =: 23; *: 4; >: 2; <: 3;
FCMP NORMAL END
READY

```

図6. パラメータ SN による出力例

## 研究 開 発

### 使用方法

#### 〔I〕 TSS

コマンドの形式

FCMP	DSN1	DSN2	PARM(SN)	
FCMPLP	DSN1	DSN2	PARM(SW)	SY(D)

#### 機能

FCMP：二つの類似のテキストファイルを比較して結果をTSS 端末へ出力する。

FCMPLP：二つの類似のテキストファイルを比較して結果をセンターのLP へ出力する。

#### パラメータ

DSN1 } 比較の対象となる二つのデータセットを指定する。  
DSN2 }

パラメータWを指定した時はDSN1で指定されたデータセットが左側、DSN2で指定されたデータセットが右側のファイルとして扱われる。

パラメータNを指定した時はDSN1で指定されたデータセットが左側の、DSN2で指定されたデータセットが右側の行番号に対応する。

PARM：パラメータはDW、DN、SW、SNの4つの組合せが可能である(2節参照)。上記の値は特に指定しない時に採用される値である。

SY：FCMPLPで出力LPの選択を行う。

D：通常のLP

A： ”

S：小文字付きLP

K：カナ文字付きLP

#### 使用例

TSSによるFCMPの使用例を図7に示す、下線を引いた部分が利用者が入力したコマンドである。

LOGON TSS F0039

+ PASSWORD ?

\*\*\*\*\*

KEQ56455I F0039 LOGON IN PROGRESS AT 13:41:21 ON AUGUST 23, 1979

JOB NO = 4227038 CENTRAL

\*\*\* WAITING FOR READY MESSAGE \*\*\*

READY

FCHP EXAMPLE.TEXT(LEFT) EXAMPLE.TEXT(RIGHT)

FACOM OSIV/F4 FCHP -790811- DATE 08/23/79 TIME 13:42:03 (N.FUJIMURA)

DSN1 DSN2 TEXT

0003\* INTEGER PRIME(1000) # SIZE OF TABLE  
\*0003 DEFINE(LIMIT,1000)

>0004 #  
>0005 INTEGER PRIME(LIMIT)

0007\* IF ( N < 1 & N > 1000)

\*0009 IF ( N < 1 | N > LIMIT) # INPUT DATA CHECK

0009\* WRITE(6,20) N; 20 FORMAT( ' INVALID N (' , I5, ' )' )

\*0011 WRITE(6,20) N; 20 FORMAT(5X, ' INVALID N (' , I6, ' )' )

0013\* PRIME(1) = 2

\*0015 PRIME(1) = 2; NO = 1; INDEX = 1; NUM = 1;

0014< NO = 1

0015< INDEX = 1

0016< NUM = 1

DSN1 = 40 LINES DSN2 = 39 LINES

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

FCHP NORMAL END

READY

TIME

KEQ56650I TIME-01:42:46 CPU-00:00:02 SESSION-00:01:24 AUGUST 23,1979

READY

TERMINAL LINESIZE(136)

READY

FCHP EXAMPLE.TEXT(LEFT) EXAMPLE.TEXT(RIGHT) PARM(SW)

FACOM OSIV/F4 FCHP -790811- DATE 08/23/79 TIME 13:43:58 (N.FUJIMURA)

DSN1 TEXT

0003 INTEGER PRIME(1000) \*\* 0003 DEFINE(LIMIT,1000) # SIZE OF TABLE

> 0004 #

> 0005 INTEGER PRIME(LIMIT)

0007 IF ( N < 1 & N > 1000) \*\* 0009 IF ( N < 1 | N > LIMIT) # INPUT DATA CHECK

0009 WRITE(6,20) N; 20 FORMAT( ' INVALID N (' , I5, ' \*\* 0011 WRITE(6,20) N; 20 FORMAT(5X, ' INVALID N (' , I6, ' )' )

0013 PRIME(1) = 2 \*\* 0015 PRIME(1) = 2; NO = 1; INDEX = 1; NUM = 1;

0014 NO = 1

0015 INDEX = 1

0016 NUM = 1

DSN1 = 40 LINES DSN2 = 39 LINES

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

FCHP NORMAL END

READY

TIME

KEQ56650I TIME-01:44:59 CPU-00:00:05 SESSION-00:03:36 AUGUST 23,1979

READY

FCHPLP EXAMPLE.TEXT(LEFT) EXAMPLE.TEXT(RIGHT)

FCHP NORMAL END ON SY(D).

READY

TIME

KEQ56650I TIME-01:45:52 CPU-00:00:07 SESSION-00:04:29 AUGUST 23,1979

READY

FCHPLP EXAMPLE.TEXT(LEFT) EXAMPLE.TEXT(RIGHT) PARM(DW)

FCHP NORMAL END ON SY(D).

READY

TIME

KEQ56650I TIME-01:46:49 CPU-00:00:10 SESSION-00:05:27 AUGUST 23,1979

READY

図7. TSSによる使用例



〔Ⅱ〕 バッチ

その機能を考えればFCMPのようなツールをバッチ処理で使用するのあまり意味がない。しかし念のため掲げる。

プロシジャ名	記 号 パ ラ メ ー タ	プロシジャステップ名
FCMP	$\text{SYSOUT} = \left\{ \begin{array}{c} \text{A} \\ \text{K} \\ \text{S} \\ \text{R} \end{array} \right\}$	FCMP

機能

二つの類似のテキストファイルを比較して指定された出力装置へ表示する。

必要なDD名

SYSUT1 } 比較の対象となる二つのデータセットを指定する。  
 SYSUT2 }

パラメータWを指定した時はSYSUT1で指定されたデータセットが左側、SYSUT2で指定されたデータセットが右側のファイルとして扱われる。

パラメータNを指定した時はSYSUT1で指定されたデータセットが左側の、SYSUT2で指定されたデータセットが右側の行番号に対応する。

記号パラメータ

SYSOUT    A    通常のLP  
           K    カナ文字付きLP  
           S    小文字付きLP  
           R    端末出力

出力形式を指定するパラメータDW, DN, SW, SNはEXEC文において、PARM.FCMPの形で直接指定する。

ジョブ制御文例

バッチ処理でFCMPを利用する場合のジョブ制御文を図8に示す。

```
//      EXEC  FCMP,PARM.FCMP=SW
//SYSUT1 DD  DSN=F0039.EXAMPLE.TEXT(LEFT),DISP=SHR
//SYSUT2 DD  DSN=F0039.EXAMPLE.TEXT(RIGHT),DISP=SHR
//
```

図8. バッチ処理のためのジョブ制御文例

## 〔Ⅲ〕制限と注意事項

1. 比較の対象となるファイルは各々9998行以下でなければならない。
2. 比較の対象となるファイルは80バイト/レコードの固定長形式に限る。
3. 各レコードの1～72欄が比較の対象となる。73～80欄は無視する。従って、例えば1欄から行番号のつくCOBOLのソーステキストをrenumberした後で比較すると問題が生じる。したがって、COBOLのソーステキストの場合は、1～6欄の行番号を除く(SAVE \* UNNUM等)などの工夫が必要である。
4. テキストはパラメータNを指定した時は1～72欄、パラメータWを指定した時は1～59欄が出力される。
5. 幅の広いTSS端末にパラメータWを指定して出力する時は、TERMINALコマンドでLINE SIZEを変更する必要がある。
6. TSSの通常のリージョンサイズでは主記憶不足が発生することがある。その時はLOGON時にSIZEパラメータでリージョンサイズを1024KB程度に大きくしてやり直す。
7. 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。