A COMPUTER SYSTEM TO MERGE MULTIPLE DATA FILES

by
J. M. Valenzuela

FOREST FIRE RESEARCH INSTITUTE
OTTAWA, ONTARIO
INFORMATION REPORT FF-X-27

CANADIAN FORESTRY SERVICE
DEPARTMENT OF FISHERIES AND FORESTRY
NOVEMBER, 1970

A COMPUTER SYSTEM TO MERGE

MULTIPLE DATA FILES

by

J. M. Valenzuela

Forest Fire Research Institute
Department of Fisheries and Forestry
Majestic Building
396 Cooper St.
Ottawa 4, Canada

Information Report FF-X-27
November 1970

TABLE OF CONTENTS

A COMPUTER SYSTEM TO MERGE

MULTIPLE DATA FILES

INTRODUCTION

Whenever large volumes of data are to be analyzed by means of a computer it is necessary to arrange the data in a form that is both suitable and efficient for computer processing. In many cases data must undergo repeated processing before it can be transformed from its original format to one which will be usable for the intended applications. It is occasionally quite simple to prepare data for computer processing, in that keypunching is often sufficient. But more often this is not the case and data must undergo much processing before it is ready for analysis by the computer.

Errors play an extremely important role in the preparatory processing of data. Undetected errors in the data or errors introduced into the data while processing can cause much havoc in subsequent steps and lead to delays and increased cost. Undetected errors could cause wrong conclusions to be reached and incorrect decisions to be made. For this reason, it is important that a system for preliminary processing of data detects as many errors as possible prior to the initiation of the analysis.

Among the various methods of preparing raw data for the initial computer processing one of the simplest and more common approaches is the following:

(a) the raw data is coded and in doing so its original computer card format is established,

(c) the data is then keypunched onto computer cards,

(c) in most cases, the keypunched data is verified in an attempt to eliminate keypunching errors.

This approach, of course, is far from being fool-proof as errors can be introduced quite easily while the data is being coded, and although the data is usually verified after keypunching, the odd keypunching error has been known to mysteriously slip through undetected. Nevertheless, the information is now on computer cards and further processing can be performed via the computer.

It is often possible to have several types of data, each type significantly different from the others. If all this data is to be analyzed simultaneously it would be more convenient if it were all merged to form a single data set. This report then describes a computer system to edit, sort and merge multiple data files.

GENERAL

The system was put into use to process data to be used by the airtanker project, which is a study being carried out by the Forest Fire Research Institute of the Department of Fisheries and Forestry to determine the feasibility of establishing a mobile airtanker force in order to assist local airtanker fleets across Canada

in fighting forest fires. Data are being collected on approximately 35,000 fires across Canada. These data are being coded and keypunched onto computer cards.

The study necesssitated the processing and reprocessing of the data a great number of times by many different computer programs. Computer cards are highly unsuitable for such repeated use both from an efficiency and economical point-of-view. They are also unsuitable for repeated operations in that with each step there is the possibility that the deck or part of it could be dropped, lost or even mixed-up with other decks and the programmer may never be aware of such an occurrence. It was decided that the most feasible way of storing the data would be on magnetic tape.

Finally, errors in the final version of the data could not be tolerated as these data would be the basis of many important decisions. All computer processing was carried out on an IBM/360-65 computer. The sorting routines used were those of the IBM system/360 Operating System Sort/Merge Program[1]. All sorting was done from tape to tape using disks for intermediate storage. The following section contains a detailed description of how the system was used to prepare the data of the airtanker project for computer processing.

INPUT

The information for each fire is contained on two computer cards, each card having a common number (fire number) located in the first five columns and a code number, a 1 for the first card and a 2 for the second, in column 80. Card-types 1 and Card-types 2 are kept in separate decks with the sequential order of the cards in each deck being unimportant.

The processing of the data as described in the example, involves three main steps and an optional fourth step:

STEP-1...editing
2...sorting
3...merging
4...sorting

At least two magnetic tapes (depending on the amount of data) are necessary, one to serve for input and the other for output while going from one step to the next.

PROCEDURE

STEP-1....Before putting the data on tape it must be carefully edited. This is accomplished by an edit program[2] written specifically to edit this particular data.

---

[1]IBM Publication No. C28-6543-3 describes the Sort-Merge program used.

[2]Edit program used in this application of the system was written by J. D. Graham, computer programmer with the Forest Fire Research Institute.

The edit program used in this example requires that the Card-type 1 deck preceed the Card-type 2 deck with a specially coded card to separate the two decks. A specially coded card is one that is significantly different from all other data cards such that the edit program will recognize it as the end of Card-type 1's and the beginning of Card-type 2's. For this application a card containing a 99999 punch in the fire no. location and a 9 in the card number code location was used. The order of the data cards within each deck is not important. The program searches for keypunching errors, faulty information due to errors in coding, missing data, duplicate data, etc. Those records found to be correct by the edit program are written on tape. The records containing errors are listed by fire number on the printer. Using this listing the errors can be corrected and new cards punched to replace the ones containing errors. The method of inserting the corrections avoids having to reprocess the entire data deck another time. Instead, only the corrected records are processed. This is done by using the edit program once again. The corrected cards are reprocessed by the edit program in the same manner as the original data decks. The only difference is that these records are written at the end of the output tape, immediately following the last record that was put on the tape on the initial run of the edit program. Corrections of Card-type 1's are followed by the specially coded card which is in turn followed by corrections of Card-type 2's. Again, the order of individual records within each deck is unimportant. Similarly, the sequence of the data on tape is not important at this stage. This operation may be repeated several times if necessary, until all the data has been corrected and is on tape.

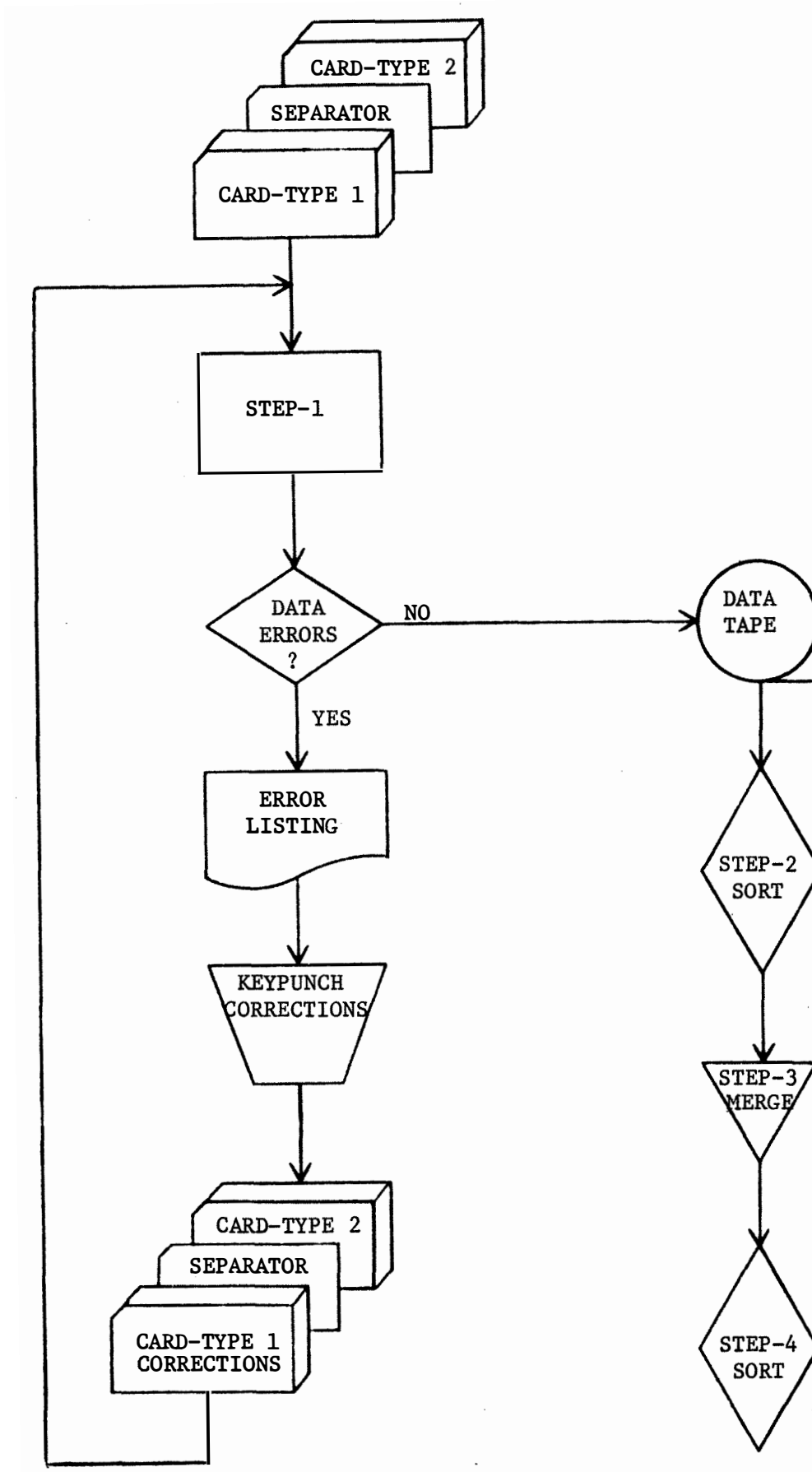
STEP-2....By this step all the data is on magnetic tape but in random order. The data is now sorted in ascending order by fire number and card number code. This is necessary to group each Card-type 1 and each Card-type 2 of common fire numbers so that they may be merged to form a single record. The statements necessary to perform this operation are listed in appendix 1.

STEP-3....As a result of STEP-2 the data is now in ascending order by fire number, with each Record-type 1 immediately followed by a Record-type 2 of the same fire number. The two records are now merged to form a single record. All unused fields are eliminated in the process. A very short Fortran program was written to accomplish the merge. It may be seen in appendix 2.

For all practical purposes this could very well be the last step as the records are all on tape and sorted in ascending order by fire number. It was desired, however, to have the final version of the data in ascending order by date within weather station. For this reason STEP-4 was necessary.

STEP-4....This step consists of a sort in ascending order of weather station and date (appendix 3).

SYSTEM FLOWCHART



```
//STEP2  EXEC  SORT
//SORTIN   DD  DSNAME=R1R2,VOL=SER=C00101,                          X
//             UNIT=2400,DISP=(OLD,KEEP),LABEL=(,NL),               X
//             DCB=(,DEN=2,RECFM=FB,LRECL=102,BLKSIZE=5100)
//SORTOUT  DD  DSNAME=R3,VOL=SER=C00122,                            X
//             UNIT=2400,DISP=(NEW,KEEP),LABEL=(,NL),               X
//             DCB=(,DEN=2,RECFM=FB,LRECL=102,BLKSIZE=5100)
//SORTWK01 DD  UNIT=2314,SPACE=,TRK,(210*,,CONTIG)
//SORTWK02 DD  UNIT=2314,SPACE=,TRK,(210*,,CONTIG)
//SORTWK03 DD  UNIT=2314,SPACE=,TRK,(210*,,CONTIG)
//SYSIN    DD *
  SORT  FIELDS=(1,5,CH,A,102,1,CH,A),SIZE=E6000
  END
/*
```

```
//STEP3        EXEC PROC=FORTGCLG,PARM.FORT='BCD'
//FORT.SYSIN  DD *
C     THIS PROGRAM MERGES AND CONDENSES REC-1 AND REC-2.
C
C     THE WHOLE RECORD WITH THE EXCEPTION OF THE FIRE NO. IS READ IN A-FORMAT
C     AS THE SOLE PURPOSE OF THE PROGRAM IS TO MERGE THE TWO RECORDS AND
C     ELIMINATE UNUSED FIELDS.
      DIMENSION A(38)
C      INPUT VARIABLE
      LR=8
C     OUTPUT VARIABLE
      LW=9
C     N TALLIES THE NO. OF RECORDS
      N=0
C     INPUT RECORD
    5 READ(LR,1,ERR=100,END=200) IFNA,(A(I),I=1,24),IFNB,(A(I),I=25,38)
     1),IFNB,(A(I),I=25,38)
    1 FORMAT (I5,24A4,/I5,13A4,A3)
C     CHECK FOR ERROR IN FIRE NO.
      IF(IFNA.NE.IFNB)GO TO 100
      N=N+1
C     OUTPUT RECORD
      WRITE(LW,2) IFNA,(A(I),I=1,38)
    2 FORMAT (I5,37A4,A3)
      GO TO 5
  100 WRITE(6,3) IFNA,IFNB
    3 FORMAT (10X,'ERROR     FIRE NO.',I6,'CARD-1     FIRE NO.',I6,'CAR
     1D-2'//' EXECUTION IS ABORTED AT THIS POINT')
      GO TO 300
C     SUCCESSFUL TERMINATION MESSAGE.
  200 WRITE(6,4) N
    4 FORMAT (//' A TOTAL OF ',I8,' RECORDS HAVE BEEN WRITTEN ON THIS TA
     1PE')
  300 CALL EXIT
      END
//GO.FT08F001 DD DSN=R3,VOL=SER=C00122,                                 X
//             UNIT=2400,DISP=(OLD,KEEP),LABEL=(,NL),                   X
//             DCB=(,DEN=2,RECFM=FB,LRECL=102,BLKSIZE=5100)
//GO.FT09F001 DD DSN=R4,VOL=SER=C00101,                                 X
//             UNIT=2400,DISP=(NEW,KEEP),LABEL=(,NL),                   X
//             DCB=(,DEN=2,RECFM=FB,LRECL=156,BLKSIZE=7800)
/*
```

```
//STEP4  EXEC  SORT
//SORTIN   DD  DSNAME=REC3,VOL=SER=C00101,                              X
//             UNIT=2400,DISP=(OLD,KEEP),LABEL=(,NL),                   X
//             DCB=(,DEN=2,RECFM=FB,LRECL=156,BLKSIZE=7800)
//SORTOUT  DD  DSNAME=R4,VOL=SER=FF0086,                                X
//             UNIT=2400,DISP=(NEW,KEEP),LABEL=(,NL),                   X
//             DCB=(,DEN=2,RECFM=FB,LRECL=156,BLKSIZE=7800)
//SORTWK01 DD  UNIT=2314,SPACE=(TRK,(210),,CONTIG)
//SORTWK02 DD  UNIT=2314,SPACE=(TRK,(210),,CONTIG)
//SORTWK03 DD  UNIT=2314,SPACE=(TRK,(210),,CONTIG)
//SYSIN    DD *
  SORT  FIELDS=(129,2,CH,A,16,6,CH,A),SIZE=3016
  END
/*
```