cc-68-37

SORT/MERGE for OS-3

by Steven K. Sullivan

September, 1968

OSU
COMPUTER CENTER
Oregon State University
Corvallis, Oregon 97331

SORT/MERGE FOR OS-3

cc-68-37

by Steve Sullivan

Computer Center
Oregon State University
September, 1968

SORT/MERGE FOR OS-3
by Steve Sullivan

1.0 USAGE

1.1 CALLING SEQUENCE OR CONTROL CARDS

The calling sequence is The Control Statement *SORT,N, where N is the logical unit number which contains the parameter cards for SORT. Logical unit 60 is used if no logical unit is specified.

1.2 ARGUMENTS, PARAMETERS

The parameter cards are used to specify what the sort program should do. Each parameter card usually contains a name followed by one or more numbers, all separated by delimiters.

All numbers are unsigned decimal numbers. The delimiters are one or more commas, equals, or blanks. In most cases, only the first letter of the name is used.

END OR ENDSORT

This parameter is the last parameter. The sort program will start sorting if no errors have been found in the parameter cards. A file mark or end of data will generate an end parameter if the end parameter is missing.

INPUT OR I

Following the word, Input or I, there may be up to ten logical unit numbers which will be used as input units in the order in which they are specified. The input units are not rewound before or after reading from them. An end of file or end of data condition will stop the sorter from reading from the current input unit. The input routine has no provision for handling labels. The users must position the input units to the start of the data. If the Input parameter is omitted, unit 60 will be used.

```
Examples:  I=4
           I=65,66,67
           I 3
           INPUT,3,4,99
```

OUTPUT OR O

Following the word, Output or O, is the output logical unit number, assumed to be 61 if omitted. The sorted output is written on this unit. The output unit is not positioned before writing. After writing, a file mark is written on the output unit. If the output unit is a file or magnetic tape, the file is positioned to before the file mark.

```
Examples:  OUTPUT = 61
           O = 73
           OUT, 3
```

KEY OR K

This parameter may be used repeatedly to specify more than one sort key. The first key is the major sort key. The last key is the least significant key. Following the word Key, or letter K, there are three numbers: the first column, the number of columns, and the type of collation desired. (A maximum number of 15,000 columns may be specified in the sort keys).

The first column number must be greater than zero and may be a column that is outside of the record. (If 80 column cards are being sorted, a person can sort on column 100). All of the cards have assumed blanks (if BCD) or assumed zeros (if binary) in columns greater than the last column of the record.

The number of columns must be greater than zero and is assumed to be one if ommitted.

The collating sequence is specified by a number between zero and seven. If omitted, it is assumed to be zero.

The meaning of the collating sequence numbers is the following:

0 - standard BCD collating sequence - Ascending Order
1 - standard BCD collating sequence - Descending Order
2 - Binary collating sequence - Ascending Order
3 - Binary collating sequence - Descending Order
4 - Signed binary collating sequence - Ascending Order
5 - Signed binary collating sequence - Descending Order
6 - Supplied with the 1st Table card
7 - Supplied with the 2nd Table card

Standard BCD collating sequence in Ascending Order

60	< 32	$ 53	(74	4 04	C 23	K 42	S 62
: 12	• 33	* 54	→ 75	5 05	D 24	L 43	T 63
= 13	) 34	↑ 55	≡ 76	6 06	E 25	M 44	U 64
≠ 14	≥ 35	↓ 56	∧ 77	7 07	F 26	N 45	V 65
≤ 15	¬ 36	> 57	0 00	8 10	G 27	O 46	W 66
% 16	; 37	/ 61	1 01	9 11	H 30	P 47	X 67
[17	- 40	] 72	2 02	A 21	I 31	Q 50	Y 70
+ 20	∨ 52	, 73	3 03	B 22	J 41	R 51	Ƶ 71

The binary and signed binary collating sequences use the CDC-3300 Internal BCD character codes to determine the collating sequence. However, the signed binary collating sequence uses the most significant bit of the field as a sign bit.

Examples: KEY 3,3,5
KEY 1,29
KEY 40 20 6

TABLE or T

The first time that the table parameter is used, table 6 is defined. Table 7 is defined the second time the table parameter is used. The table parameter may be used only twice. After the word table, or letter t, there should be one delimiter followed by a string of BCD characters. The string of characters is terminated with the second occurence of any BCD character. The string of characters determines the collating sequence. All characters not in the string will be sorted last.

Examples: 1) A collating sequence to order playing cards could be defined by the following table parameter: TABLE= AKQJT987654322

2) To order pinocle playing cards the following table parameter could be used: TABLE,ATKQJ9A

3) To order playing cards by suits, the following table may be used: TABLE..SHDCC

RECORD OR R

The record length (in words) should follow the word RECORD or letter R. If variable length records are to be sorted (no blocking or un-blocking) the record length should be equal to or greater than the longest record. Records longer than the specified record length will be truncated to the record length. For blocked records the correct record length must be specified with this parameter. Both the sort phase and the merge phase become more efficient as the record length parameter becomes smaller; however, the record length may usually be as large as 10,000 words.

```
Examples:  R 20
           RECORD.LENGTH=250
```

PARITY OR P

Following the word Parity, or letter P, should be a zero, one, or two. If the parity parameter is omitted, or if the digit is a zero, sort will stop on an irrecoverable parity error. If the digit is a one, all records or blocks of records with irrecoverable parity errors will be skipped. If the digit is a two, all records or blocks of records with irrecoverable parity errors will be used. In any case an error message will be written on unit 61 for each irrecoverable parity error.

```
     Examples:  P=1
                PARITY, 2
                P 2
```

BI (BLOCKED INPUT)

Following the letters BI is a number which specifies the maximum number of logical records in any input block. All input blocks must be a multiple of the logical record length and may not contain more than BI logical records in any block. Should either of these conditions not be met, sort will stop after printing an error message.

Example: BI=100

BO (Blocked Output)

Following the letters BO is a number which specifies the maximum number of logical records in any output block. All output blocks will be the maximum length except the last block, which may be less than or equal to the maximum length. All output blocks will be a multiple of the logical record length even if it is necessary to lengthen input records (blank or zero fill short input records.)

Example: BO=100

1.3 CORE REQUIREMENTS

Core memory is dynamically allocated and allocated differently at different times. Before sorting, the sort program, which is about 1250 decimal words long, is loaded into the high end of memory (77777). During sorting the sort program is about 900 decimal words long, releasing about 350 more words to be used as scratch. During merging, only about 540 decimal words of the sort program are used.

During sorting, core is used as follows, starting at location 20:

1) 508_{10} words are used as a merge output buffer.
2) 32_{10} words are used to contain the merge output logical unit numbers and flags for merge output.
3) 32_{10} words are used to contain the number of output strings on each merge output unit.
4) One word for every column in every sort key plus one word is used to store information about the sort keys.
5) One word for every column in every sort key plus two words is used to store information while writing sorted strings.
6) A fixed length input buffer the size of input blocks is used only if the BI parameter was used.
7) A fixed length output buffer, the size of the output blocks, is used only if the BO parameter was used and only for the first sorted string. (Sorted strings are written on a merge unit and an output buffer is not needed. However, if no merging is necessary, the output buffer would be used).
8) All remaining storage is used by records being sorted and pointers. Each N-word logical record uses N+3 words plus 2(p-q) words where p is the number of characters in the sorted portion of the record necessary to make the record unique. Q is the number of characters in the most similar record necessary to make it unique.

Example: Assume key 1, 10 and three records THEX,THIS, THERE.

Input THEX. Both p and q = 0
Input THIS. It takes three characters to make THIS unique. p=3, q=0. Input THERE. It takes four characters to make THERE unique. However, it takes three characters to make THEX unique from THERE. p=4, q=3, p-q=1.

During merging, core is used as follows:

1) same as 1) for sort
2) same as 2) for sort

3) same as 3) for sort
4) same as 4) for sort
5) same as 7) for sort
6) 93_{10} words are used as pointers for the tournament sort.
7) 32_{10} words are used to contain the merge input logical unit numbers and flags.
8) 32_{10} words are used to contain the number of remaining merge input string on each merge input unit.
9) K([M/4] + 2) word for storing information is used to sort the records. K is the number of merge units and M is the number of columns in the sort keys.
(D)K(N+1) words are used for storage of records where N is the maximum number of words in each record.
(1)K(509) words are used for merge input buffers.

K is evaluated by calculating ([m/4]+2+509+N+1) and dividing it into the number of available words. K cannot be greater than 32 or less than two.

1.5 FORMATS

Input to be sorted should be either all binary or all BCD. The first record is tested to determine if it is BCD or binary. The resultant mode is used for all further reads. If the first record was BCD, the record is assumed to have trailing blanks; if binary, trailing zeros are assumed.

1.6 ERROR MESSAGES

PARAMETER ERROR

Here are some of the things that can cause this error:

1) words that start with letters or symbols other than B, E, I, K, O, P, R, or T.
2) More than ten input units specified.
3) More than one output unit

4) Input or output units greater than 99.

5) The third parameter after KEY being greater than 7.

6) The sum of the first and second parameter after KEY being greater than 15,000

7) More than two TABLE parameters. If this error occurs after the ENDSORT parameter, the error is probably one of the following:

8) No KEY parameter.

9) The available storage is not large enough for two merge units. ((BO)*(R)+(2)*(R)+(1.5)*(N)>30,000 where R is maximum record site, and N is the number of columns in all sort keys.)

PARITY ERROR ON LUN XX

An irrecoverable parity error was found on logical unit XX.

BLOCK LENGTH ERR ON LUN XX

A record read from logical unit XX either was longer than (R)*(BI) words or was not a multiple of (R). (Unit XX may be backspaced and read by the user).

TOO MANY LUNS EQUIPPED

During sorting and merging, a maximum of 64 internal files (files not equipped by the user) may be used. There are a maximum of 100 files, leaving at least 36 for the user.

1.7 TIMING

No thorough study of timing has been made. However, it is possible to make some crude estimates of timing using the following formula: T= [(250vs)w+ (llms)]R

w is number of words in each record, r is the number of records to be sorted, t is CPU time.

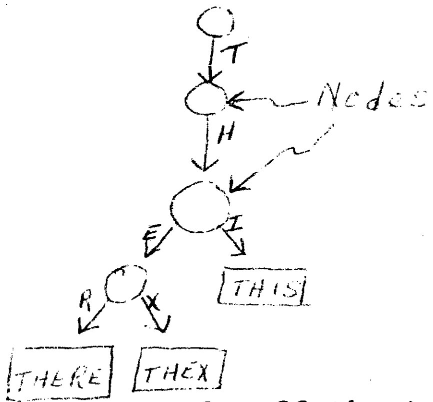
For sorts of less than about 1,000 cards, no merge passes will be necessary and CPU times may be better than the formula indicates. For sorts of more than about 32,000 cards, more than one merge pass will be necessary and times may be longer than the formula indicates.

Wallclock time should be about three times longer than CPU time.

2.0 METHOD OF ALGORITHM

Sorting

During sorting, records are read into an ordered data structure. The data structure is such that the records can be written out in correct order. The data structure is a tree with ordered nodes for every significant character in every record. The tree structure for THEX, THIS, and THERE would be:



By "picking" the records off the tree from left to right, the records will be in order.

If memory becomes full while sorting, all records read are written into a merge unit. Up to 32 merge units may be used.

Merging

If memory does not become full, the output from sort goes directly to the output unit. If merging is necessary, merge reads from all the previous merge output units into as many as 32 more merge units. During merging, great care is taken to keep pre-ordered records from being shuffled when information in the sort keys is identical. Each time a merge pass is completed, the sorted strings on the merge units become 32 times longer. The final merge pass merges from up to 32 units into the output unit. All merge files that sort and merge had equipped are unequipped. A filemark is written on the output unit and the output unit is backspaced if it is a file or magnetic tape.

Tournament replacement sorting is done during merging, using multi-level indirecting and masked equality searches.

3.0 SAMPLE PROBLEMS

1) A magnetic tape is to be sorted and the output written into the line printer. The tape contains 932 records that are 120 characters long. The records are not blocked. The first 25 characters are to be sorted using the standard collating sequence.

```
7  JOB,xxxx,xxx
8
7  EQUIP,1=MT 7304 AT 556
8
7  *SORT
8
   I 1
   K 1 25
    END
   [output is listed here]
7  LOGOFF
8
      TIME 17.300 SECONDS MFBLKS 119
```

2) A magnetic tape contains information about files used under an operating system. Columns 4 through 8 and 11 through 16 contain the job-user numbers. Columns 21 through 28 contain the file name and columns 37,38,40,41,43,44,47, and 48 contain the month, day, year and hour of access.

It is necessary to list the information on the tape in such a way that all files that were accessed during a particular hour be listed by job-user number and then file name. It is also necessary to use a special collating sequence on the file name.

```
                                      7 JOE,xxxx,xx
                                      8
                                      7 EQUIP, 1=MT 7305
                                      8
                                      7 *SORT
                                      8
Read from unit 1, write on 61         I 1 OUTPUT 61
Sort first on year, month             K 43 2 K 37 2
Sort on day and hour                  K 40 2 K 47 2
Sort on job number                    K 4 5
Sort on user number                   K 11 6
Sort on file name                     KEY 21 8 6
Specify file name collating sequence  TABLE␣␣*0123456789ABCD.....
                                      ENDSORT
                                      [output is here  2,418 records.]
                                      7 LOGOFF
                                      8  TIME 37.200 SECONDS
                                         MFBLKS 161
```